

**MODÈLE SÉMANTIQUE D'INTELLIGENCE AMBIANTE
POUR LE DÉVELOPPEMENT *DO-IT-YOURSELF*
D'HABITATS INTELLIGENTS**

par

Hubert Kenfack Ngankam

Thèse présentée au Département d'informatique
en vue de l'obtention du grade de philosophiæ doctor (Ph.D.)

FACULTÉ DES SCIENCES
UNIVERSITÉ DE SHERBROOKE

Sherbrooke, Québec, Canada, 16 janvier 2019

Le 16 janvier 2019

Le jury a accepté la thèse de Monsieur Hubert Kenfack Ngankam dans sa version finale.

Membres du jury

Professeur Sylvain Giroux
Directeur de recherche
Département d'informatique

Professeur Hélène Pigot
Co-Directrice de recherche
Département d'informatique

Emmanuel Conchon
Maître de Conférences
Membre externe
Département d'informatique
Université de Limoges

Charles Gouin-Vallerand
Professeur
Membre externe
Département Science et Technologie
Université TÉLUQ

Professeur Gabriel Girad
Président-rapporteur
Département d'informatique

Sommaire

L'intelligence ambiante utilise l'informatique diffuse, l'Internet des objets et l'intelligence artificielle pour faire émerger de nouvelles solutions technologiques qui facilitent le maintien à domicile des personnes âgées. L'assistance à l'autonomie à domicile passe par un processus de transformation de l'habitat en habitat intelligent.

Cette thèse s'intéresse à la proposition d'un modèle sémantique de l'intelligence ambiante. Le but est de proposer un domaine d'interprétation des interactions existantes entre l'habitat, l'Internet des objets, le profil de la personne et l'assistance à offrir. À ce titre, le lien sémantique entre les énoncés permet de formellement spécifier les besoins et les règles d'inférences sont utilisées pour la génération des dispositifs électroniques correspondant aux besoins spécifiés.

Le modèle sémantique proposé capture l'essence des concepts du domaine de l'intelligence ambiante. Il organise et structure ainsi la connaissance sous-jacente au sein d'une architecture logicielle. Couche par couche et module par module, cette architecture s'adapte au contexte de l'utilisateur afin de permettre aux personnes ayant peu ou pas de connaissances dans le domaine de concevoir (bricoler) leur propre solution d'assistance et d'accompagnement à la réalisation de certaines activités de la vie quotidienne.

Pour faciliter le bricolage (la conception par soi-même), le modèle s'appuie sur des méthodes formelles de spécification, de vérification et de validation des scénarios. Les scénarios décrivent l'agencement et le déroulement des activités à réaliser pour la satisfaction d'un but. Enfin, pour accompagner le processus de bricolage jusqu'au bout, un modèle théorique de positionnement des dispositifs électroniques dans un habitat intelligent est proposé.

Une expérimentation à domicile a été menée durant 42 jours chez une personne

SOMMAIRE

aînée souffrant de la maladie d'Alzheimer. Les résultats obtenus permettent de valider le fonctionnement du modèle de sensibilité au contexte proposé et de renseigner les proches aidant sur les habitudes de vie de leur aînée. L'assistance offerte a pour objectif de fournir des signaux environnementaux pour favoriser un sommeil réparateur et éviter des épisodes d'errance nocturne, chez les personnes atteintes de la maladie d'Alzheimer.

Mots-clés: Intelligence Ambiante ; Sensibilité au Contexte ; Informatique diffuse ubiquitaire ; Habitats intelligents ; Ontologie ; Scénarios ; Spécification formelle ; Do-It-YourSelf ; Ingénierie des connaissances ; Programmation linéaire ; Modèles

Dédicace

À vous : Ochriel, Kaleb & Owen ...

Remerciements

Comment pourrais-je commencer ce travail sans vous dire merci ? Votre soutien scientifique, pédagogique et moral a permis que je trouve ma voie. De manière agréable et sympathique, vous m’avez toujours montré le chemin en m’aidant à l’arpenter. La confiance que vous avez placée en moi et la liberté offerte d’explorer plusieurs aspects du même problème m’ont rendu éveillé et libre. Donc je le dis pour ne pas l’oublier, merci à toi, Hélène Pigot et à toi Sylvain Giroux, mes directeurs de thèses sans qui ce travail n’aurait jamais vu le jour.

Je tiens aussi à remercier Marc Frappier pour sa collaboration scientifique à qui j’associe Jean Goulet pour l’encadrement pédagogique et pour la gestion de mon plan de formation.

Je voudrais aussi remercier mon jury de thèse pour l’évaluation, les suggestions et les commentaires qu’ils apportent à l’amélioration de ce travail.

Merci à AGE-WELL NCE pour le financement accordé pour cette thèse et pour tout l’effort mis pour nous permettre d’acquérir des connaissances transversales et pluridisciplinaires.

Je remercie les membres de l’équipe du projet « AGE-WELL » pour tous les beaux moments de sprints, d’issues, de bogues, de stories, bref d’agilité et de travail d’équipe, merci Marie-Michèle Rosa-Fortin, Pierre-Yves Groussard, Robert Radziszewski Maxime Parenteau, Damien Lockner, Camilar Olivera, Yannick Drolet, Léopold Lieb, Armel Ayimdji et Corentin Haidon.

Je voudrais également remercier Isabelle Viens et Dominique Lorrain pour leur accompagnement lors des expérimentations à domicile, les attentions et le regard psychologique qu’elles ont apporté au scénario d’errance nocturne.

À vous Domus ! Oui toi, Frédéric Bergeron, Wathek Bellah Loued, Mathieu Ga-

REMERCIEMENTS

gnon, Jules Randolph, Catherine Laliberté, Célia Lignon, Ariane Tessier, Hélène Benoît, je voudrais vous dire merci pour l'ambiance, pour l'attention, pour l'accueil au sein du laboratoire et pour l'écoute attentive que vous m'avez accordée durant ces belles années. Merci aussi pour les parties de jeu de tarot.

À tous les enseignants du département de génie informatique de l'école nationale supérieure polytechnique de Yaoundé au Cameroun, je voudrais vous témoigner toute ma sympathie et la reconnaissance pour toutes les années passées à vos côtés.

Merci à toi, Claude Marie Ngabireng, j'ai adoré être à tes côtés durant ces années, tu m'as tellement fait confiance que j'ai compris le vrai sens des responsabilités.

Toi Georges Édouard Kouamou, merci d'avoir eu confiance en moi, de m'avoir soutenu et encadré durant tous ces moments difficiles. Tes appels m'ont toujours apporté de l'espoir. Nous l'avons construit ce viaduc.

Merci à toi, Alliot Tamouya, durant longtemps tu m'as forgé doucement, mais profondément, lentement mais sûrement.

A toi, Félix Fomo, pour toute la confiance et l'encadrement fraternel, je voudrais te témoigner ma gratitude.

Merci à la famille Fomo, j'ai toujours progressé grâce à vous, car vos conseils respectifs ont toujours su satisfaire mes attentes. La preuve, je suis avec vous pour toujours.

Merci à tous mes enfants Ngankam : Franck, Dalina, Ivana, Noé, Péniele, Grâce, Edena, Kyliane, Evan, Yohan, Brady, Gadiela, Maya, Candice, Andres, Joaquim, Ochriel, Kaleb, Owen et Adiel pour les magnifiques sourires que vous m'offrez à chacune de nos rencontres.

Merci à toutes mes femmes : Rosine, Alida, Arlette, Laura, Josiane, Anne-Michèle vous qui avez toujours su me comprendre malgré mes manquements.

Merci à mes frères : Guy, Sylvain, Jacques, Thierry et Charles pour l'amour et la confiance inconditionnelle.

Merci à ma grande sœur chérie, Virginie, qui m'a toujours tenu par le bras.

À ma très chère et tendre épouse, ma muse, mon égérie et ma plus grande victoire Josiane Djuko, merci d'avoir toujours été présente et d'avoir mis à ma disposition une écoute intelligente.

À toi, maman Rose Ngankam, cette thèse est ta plus grande victoire, bravo !

Abréviations

AGE-WELL NCE Aging Gracefully across Environments using Technology to Support Wellness, Engagement and Long Life NCE Inc., page 2

AmI Ambient Intelligence, page 30

API Application Programming Interface., page 5

AVQ Activités de la Vie Quotidienne, page 1

CIUSSS Centre Intégré Universitaire de Santé et de Services Sociaux, page 136

CTT Concur Task Trees, page 51

DIY-AIDE Do-It-Yourself Adaptable Intelligent Domestic Environments., page 2

DIY Do-It-Yourself., page 5

DOMUS Domotique et informatique mobile à l'Université de Sherbrooke, page 1

DRS Dementia Rating Scale, page 136

HTA Hierarchical Task Analysis, page 51

MoCA Montréal Cognitive Assessment, page 136

MQTT Message Queuing Telemetry Transport, page 114

ABRÉVIATIONS

- NEARS** iNnovative Easy Assistance Reassuring System., page 2
- OWL** Ontology Web Language, page 21
- RDF** Resource Description Framework, page 20
- REST** REpresentational State Transfer, page 100
- SAT** SATISFIABILITY, page 71
- SPARQL** SPARQL Protocol and RDF Query Language, page 38
- SWRL** Semantic Web Rule Language, page 118
- UML** Unified Modeling Language., page 5
- URI** Uniform Resource Identifier, page 20
- W3C** World Wide Web Consortium, page 20
- XML** eXtensible Markup Language, page 20

Table des matières

Sommaire	ii
Dédicace	iv
Remerciements	v
Abréviations	vii
Table des matières	ix
Liste des figures	xiv
Liste des tableaux	xvii
Introduction	1
I État de l’art	11
1 Intelligence Ambiante, Sensibilité au Contexte, DIY & Ontologie	12
1.1 L’informatique diffuse	13
1.1.1 Contexte et définition	13
1.1.2 Informatique ubiquitaire – Technologie calme et DIY	14
1.2 La modélisation du contexte	15
1.2.1 Le contexte	15
1.2.2 La modélisation du contexte	16
1.2.3 La sensibilité du contexte	17

TABLE DES MATIÈRES

1.3	Ontologie	19
1.3.1	Données – Informations – Connaissances	19
1.3.2	Les ontologies	20
1.3.3	Langage et formalisme	22
1.4	Spécifications formelles	23
1.4.1	Spécification et assistance aux personnes âgées	23
1.4.2	Sémantique - Vérification - Raffinement	24
1.4.3	Les outils de spécifications	25
1.5	Do-it-Yourself	26
1.5.1	Contexte et définition	26
1.5.2	DIY et environnements intelligents	28
1.6	Intelligence ambiante	30
1.6.1	Contexte et définition	30
1.6.2	Les domaines des systèmes d'intelligence ambiante	32
1.6.3	Les domaines d'applications de l'intelligence ambiante	33
1.6.4	L'intelligence ambiante dans la littérature	34
1.7	Synthèse de l'état de l'art	39
1.8	Conclusion	40
2	Les Habitats Intelligents	41
2.1	Des habitats automatiques aux habitats intelligents	41
2.2	Les technologies pour les habitats intelligents	43
2.3	Les solutions d'habitats intelligents existantes	44
2.3.1	Les solutions de laboratoires de recherche	44
2.3.2	Les solutions industrielles	47
2.4	Conclusion du chapitre	48
II	Contributions	49
3	Spécification Formelle de Scénarios	50
3.1	Définition des concepts clés	51
3.1.1	Définition : Scénario	51

TABLE DES MATIÈRES

3.1.2	Définition : Scénario d'assistance	51
3.1.3	Définition : Activité	52
3.1.4	Définition de tâches	52
3.1.5	Définition d'actions	52
3.1.6	Définition de conception dirigée par les scénarios	52
3.2	Des exigences multidisciplinaires aux scénarios	53
3.3	Caractéristiques du modèle hiérarchique	55
3.4	Exemples de scénarios	56
3.4.1	Scénario 1 : Aller aux toilettes durant la nuit depuis sa chambre à coucher	56
3.4.2	Scénario 2 : Boire un verre d'eau au robinet dans la cuisine . .	56
3.4.3	Scénario 3 : Se relaxer au salon	57
3.4.4	Scénario 4 : Errance nocturne	59
3.5	Modélisation	59
3.5.1	Approche ascendante (Bottom-up Approach)	63
3.5.2	Approche descendante (Top Down Approach)	64
3.5.3	Approche par combinaison	64
3.6	Formalisme	66
3.6.1	Le choix de ALLOY comme outil de spécifications formelles . .	67
3.6.2	Du Scénario à la spécification formelle en ALLOY	67
3.6.3	La spécification de contraintes	68
3.7	Implémentation d'une instance de la spécification : cas du scénario d'errance nocturne	71
3.8	Résultats de la validation et de la spécification de scénarios	74
3.8.1	Résultats de la spécification du scénario 1	74
3.8.2	Résultats de la spécification du scénario 2	76
3.8.3	Résultats de la spécification du scénario 3	78
3.8.4	Résultats de la spécification du scénario 4	79
3.9	Conclusion du chapitre	79
4	Modèle Sémantique des Systèmes Ambiants	82
4.1	De la donnée à la connaissance sémantique multiniveaux	82

TABLE DES MATIÈRES

4.2	La conceptualisation de l'ontologie	85
4.3	L'implémentation de l'ontologie	87
4.3.1	Modélisation de l'environnement	90
4.3.2	Modélisation des scénarios	93
4.3.3	Modélisation des capteurs/effecteurs	96
4.4	Conclusion partielle	99
5	Implémentation du Modèle Sémantique	100
5.1	L'implémentation du modèle	100
5.2	Implémentation de l'API	102
5.2.1	Hétérogénéité des données et des applications	102
5.2.2	La conception par des procédés de méthodes agiles	102
5.2.3	Implémentation sémantique	104
5.3	Conclusion du chapitre	105
6	Architecture d'Exécution du Contexte décrit dans le Modèle Ontologique	106
6.1	Pourquoi être sensible au contexte	107
6.2	Exemple d'application	107
6.3	Modèle architectural de sensibilité au contexte	108
6.3.1	Le Module de collecte de données	109
6.3.2	Le Module de traitement et d'adaptation de contexte	115
6.3.3	Le Module de gestionnaire de contexte	118
6.4	Conclusion partielle du chapitre	119
7	Modèle de Positionnement des Capteurs	121
7.1	La problématique de positionnement des capteurs	121
7.2	Le Modèle de positionnement	122
7.2.1	Problème de positionnement	122
7.2.2	Définition d'un problème d'optimisation combinatoire	123
7.3	La modélisation du problème	123
7.3.1	Positionnement pour une détection en 2D	124
7.3.2	Positionnement pour une détection en 3D	126

TABLE DES MATIÈRES

7.4	Spécification du modèle	126
7.5	Exemple de positionnement de capteurs : le cas du Laboratoire Domus	129
7.5.1	Description de la pièce	129
7.6	Conclusion du chapitre	133
8	Expérimentation et Résultats	135
8.1	Démarche expérimentale	135
8.1.1	Profil de la participante	135
8.1.2	Protocole expérimental	136
8.1.3	Calendrier de l'expérimentation	138
8.1.4	Résultat du test cognitif de Pauline	138
8.2	De l'habitat à l'habitat intelligent de Pauline	140
8.3	Matériels et méthodes	142
8.4	Discussion sur les résultats	144
8.4.1	Évaluation de l'objectif : Architecture de sensibilité au contexte	144
8.4.2	Évaluation de l'objectif : de collecte de données	148
8.4.3	Évaluation de l'objectif : l'assistance offerte à Mme Pauline . .	152
8.5	Conclusion	156
	Conclusion	157
A	Les 8 principes de la technologie calme	164
B	Manifeste DIY pour L'Internet des objets	166
C	Les instances d'exécution en ALLOY du scénario errance nocturne	168

Liste des figures

1.1	Architecture du Context Toolkit components [33]	18
1.2	Typologie de création DIY dans l'Internet des objets [93]	29
1.3	Évolution de l'informatique vers l'intelligence ambiante de Waldner 2007	31
1.4	Hierarchie ubiquitaire - Sensibilité au contexte – Intelligence Ambiante	33
3.1	Processus standard pour la production des spécifications utilisateurs [65]	53
3.2	Processus multidisciplinaires de co-conception hiérarchique de scénarios [65]	54
3.3	Utilisation du varioglass	58
3.4	Exemple d'indicateur lumineux	58
3.5	Approche ascendante de création de scénarios	64
3.6	Approche descendante pour la création de scénarios	65
3.7	Combinaison de scénarios : création d'un nouveau scénario	65
3.8	Combinaison sans création d'un nouveau scénario	66
3.9	Exemple de spécification de base d'un scénario en ALLOY	68
3.10	Exemple de spécification de base d'un scénario en ALLOY	71
3.11	Exemple de spécification de base d'un scénario en ALLOY	73
3.12	Représentation hiérarchique du scénario : aller aux toilettes	76
3.13	Représentation hiérarchique du scénario : «boire de l'eau à la cuisine»	77
3.14	Représentation hiérarchique du scénario : «se relaxer au salon»	79
3.15	Représentation hiérarchique du scénario : «errance nocturne»	80
4.1	Intégration de la connaissance dans le raisonnement	84
4.2	Étape de développement de l'ontologie	85

LISTE DES FIGURES

4.3	Vocabulaires et questions autour d'une assistance ambiante pour le cas « sortir du lit »	88
4.4	Vue globale de l'ontologie	89
4.5	Résultat de la modélisation du cas d'utilisation « sortir du lit » . . .	91
4.6	Vue partielle de l'ontologie du domaine <i>Home</i>	92
4.7	Spécifications formelles de la définition du concept de pièce	93
4.8	Vue partielle de l'ontologie du domaine <i>Task</i>	94
4.9	Définition formelle d'un scénario	95
4.10	Définitions formelles d'une tâche	95
4.11	Vue partielle de l'ontologie de domaine <i>Devices</i>	97
4.12	Autre vue partielle de l'ontologie de domaine <i>Devices</i>	98
4.13	Définitions formelles d'un contrôleur	98
5.1	Architecture fonctionnelle du modèle sémantique	101
5.2	Exemples de format de données JSON pour un scénario	103
5.3	Exemple de requête SPARQL pour le choix des tâches	104
6.1	Diagramme d'activité de Georges	109
6.2	Architecture logicielle pour les applications sensibles au contexte . . .	110
6.3	Environnements d'acquisition de contexte dans un habitat intelligent	112
6.4	Architecture physique du modèle de sensibilité au contexte	113
7.1	Définition formelle et générale d'un programme linéaire	123
7.2	Exemple de chevauchement de capteurs	124
7.3	Positionnement des capteurs selon la version ellipse 2D	125
7.4	Positionnement des capteurs selon la version en cônes 3D	126
7.5	Zone pour le positionnement de capteurs de mouvement	130
7.6	Installation du capteur de mouvement au plafond : (source manuel officiel)	131
7.7	Installation du capteur de mouvement au mur : (source manuel officiel)	133
8.1	La SMARTDOMUS de Pauline	141
8.2	Modification d'un capteur d'eau pour le transformer en capteur de pression	142

LISTE DES FIGURES

8.3	Total du temps passé par nuit et par pièce entre 20 heures et 8 heures.	149
8.4	Temps passé hors de la chambre à coucher par nuit.	150
8.5	Classification des nuits par regroupement d'heures avec le nombre d'instances par classes	151
8.6	Zoom sur les activités par pièce et par minute pour la 30 ^e nuit	152
8.7	Zoom sur les activités par pièce et par minute pour la 34 ^e nuit	153
8.8	Zoom sur les activités par pièce et par minute pour la 37 ^e nuit	154
8.9	Comparaison des activités dans les pièces avec et sans effecteurs	155
8.10	Activation des effecteurs par nuit et par levée	155
B.1	Manifeste DIY pour L'Internet des objets [31]	167
C.1	Exemple d'instances pour la première itération du scénario errance nocturne	169
C.2	Exemple d'instances pour une ième itération du scénario errance nocturne	169

Liste des tableaux

3.1	Résultats de l'exécution de l'assertion de la spécification de base . . .	74
3.2	Résultats de la vérification des assertions pour le scénario 1.	75
3.3	Résultats de la vérification des assertions pour le scénario 2	78
3.4	Résultats de la vérification des assertions pour le scénario 3	78
3.5	Résultats de la vérification des assertions pour le scénario 4	80
8.1	Calendrier de l'expérimentation	139
8.2	Liste des équipements utilisés chez Pauline.	143
8.3	Caractéristique de Pauline.	146
8.4	Résultats de la classification des nuits par K-means	152

Introduction

Contexte

Te rappelles-tu ? Comprends-tu ? T'en souviens-tu ? Ou préfères-tu que je te laisse te reposer ? Je ne te dirais pas quoi faire, je ne te montrerais pas comment le faire. Toute ta vie durant, tu savais quoi faire, comment le faire et quand le faire. Toutes ces années sont une expérience incalculable. Comme le dit un proverbe africain : « Dans l'œil des personnes âgées se trouve le chemin de la vie. » Je voudrais donc pour tout le reste du temps être près de toi quand tu es seule. Être une extension de ta mémoire quand tu oublies, t'accompagner de manière sécuritaire quand tu es anxieuse, quand tu es perdue et quand tu as besoin de moi. Rappelle-toi que c'est une grâce d'avoir vécu aussi pleinement ta jeunesse, alors pourquoi faudrait-il que ta vieillesse soit si hasardeuse ? Surtout, n'oublie pas que des millions de personnes sont dans la même situation que toi. De plus, une étude de l'OMS montre qu'ils seront environ un milliard à te rejoindre.

Le laboratoire DOMUS (Domotique et informatique mobile à l'Université de Sherbrooke) l'a bien compris et se positionne donc comme cette personne qui accompagne en montrant le chemin aux personnes âgées. Depuis 2002, il utilise des technologies d'information et de la communication pour faciliter le maintien à domicile aux aînées et aux personnes ayant des troubles cognitifs. Pour faciliter l'acceptation des technologies comme appui à la réalisation des AVQ (Activités de la Vie Quotidienne) dans le quotidien des aînés, le DOMUS utilise des procédés innovants et des approches multidisciplinaires centrés utilisateurs.

Aujourd'hui grâce aux années d'expérience et aux multiples travaux et aux expérimentations à domicile effectués, le laboratoire a compris qu'il fallait changer de

INTRODUCTION

paradigme, qu'il fallait aller plus loin en donnant plus d'autonomie, plus de pouvoir, plus de liberté et plus de possibilités à l'utilisateur final pour innover. La volonté sous-jacente est de déporter le laboratoire au domicile pour le rendre intelligent, et mieux encore, de donner la possibilité aux personnes ayant peu ou pas de connaissances dans le domaine de rendre leur habitat intelligent.

En 2013, le laboratoire porte ce projet d'habitat intelligent par les utilisateurs auprès de AGE-WELL NCE (Aging Gracefully across Environments using Technology to Support Wellness, Engagement and Long Life NCE Inc.) , un réseau canadien favorisant l'utilisation des technologies pour un meilleur vieillissement à domicile. Ce réseau de centre d'excellence vise à aider les Canadiens âgés à conserver leur indépendance, leur santé et leur qualité de vie grâce à l'utilisation des technologies et des services qui accroissent leur sécurité. En accompagnant, les aidants naturels, il promeut par là le soutien à une vie autonome et aux renforcements de la participation sociale des aînés. Le laboratoire a donc reçu de AGE-WELL le support pour conduire des travaux sur les habitats intelligents construits par les aidants naturels.

Pour prendre en compte la complexité multi échelle et offrir un prototype fonctionnel à 360 degrés, le laboratoire s'est associé à d'autres laboratoires dans plusieurs universités du Canada. C'est ainsi que le projet DIY-AIDE (Do-It-Yourself Adaptable Intelligent Domestic Environments) a vu le jour. Pour mieux gérer les spécificités, le projet a été scindé en plusieurs sous-projets. NEARS est la déclinaison actuelle de la partie du projet qui est en cours de développement au sein du laboratoire DOMUS. NEARS (iNnovative Easy Assistance Reassuring System) est une intelligence artificielle permettant aux aînés de vivre de manière autonome chez eux. Pour parvenir à rendre intelligent l'habitat de ceux qui l'utilisent, NEARS se propose de cibler des activités de la vie quotidienne (AVQ) auxquelles apporter une assistance, de concevoir un système automatique de sélection, d'installation et de gestion des dispositifs électroniques nécessaires à la compréhension des besoins de la personne. Le but est de permettre que l'ensemble s'exécute parfaitement, en s'auto-gérant et s'auto-réparant.

Cette thèse s'inscrit dans la seconde partie c'est-à-dire la conception du système automatique de sélection, d'installation et de gestion des dispositifs électroniques (capteurs, effecteurs...). Ce travail puise donc sa source à la fois dans la modélisation sémantique, l'ingénierie des connaissances et la spécification formelle pour y apporter

INTRODUCTION

une solution. Concrètement, il s'agit de mettre en place la structure architecturale de sensibilité au contexte pour permettre à chaque utilisateur de concevoir une assistance ambiante sur mesure pour la réalisation de ses besoins ou de celui de son aîné.

Problématique

Les habitats intelligents par définition sont doués de facultés d'adaptation, de perception et de raisonnement. Ils ont pour but d'apporter de la valeur ajoutée au bien-être des personnes de manière calme et non intrusive. Les solutions existantes n'offrent pas la possibilité à l'utilisateur final de concevoir sa propre solution. Néanmoins, elles offrent aux utilisateurs des niveaux de configuration et de paramétrage très évolués, sans pour autant leur offrir la capacité d'innover ou de bricoler. Les utilisateurs dans ce cas sont des consommateurs passifs, limités dans leur choix par des concepteurs et des développeurs malgré, les approches centrées utilisateurs utilisées. Nous nous proposons de faire des utilisateurs des bâtisseurs, des bricoleurs, des façonneurs, des designers et des *consom'acteur* (un consommateur qui n'accepte plus passivement les biens et les services qu'on lui propose.) de l'intelligence ambiante de leur habitat.

Cette thèse s'inscrit dans la modélisation sémantique des connaissances pour rendre une application sensible au contexte. De manière orthogonale, ces travaux permettraient la mise en place d'un écosystème qui faciliterait aux personnes ayant peu ou pas de connaissances en Internet des objets de créer/concevoir des solutions d'assistances ambiantes.

En d'autres termes, il s'agit pour nous de mettre en place, une démarche formelle pour permettre à l'utilisateur final de partir d'un besoin quelconque de très haut niveau (aller aux toilettes) pour valider la cohérence et la conformité du besoin afin de générer la liste et la position des capteurs dans son habitat. Le besoin utilisateur peut être un simple besoin (ouvrir une porte) ou un besoin plus complexe (limiter l'errance nocturne, préparer un repas. . .). Il nous revient donc de proposer une solution permettant de générer des spécifications formelles pour la création des scénarios d'assistance afin d'assurer la conformité et la validation des besoins de l'utilisateur. Cette solution aura pour but de générer les listes des effecteurs associées ainsi que leur

INTRODUCTION

emplacement dans le domicile. Plus simplement, la question se pose en ces termes :

Quels dispositifs de l’Internet des objets (capteurs, effecteurs, contrôleurs, téléphones...) doit-on choisir pour accompagner les personnes dans la réalisation de leurs scénarios d’assistance ?

Le modèle sémantique développé met en symbiose l’environnement physique, les capteurs/effecteurs et le scénario validé pour produire le nombre de capteurs/effecteurs et la position à occuper par chacun d’eux. Il doit permettre d’une part, l’expression, la définition, la structuration et l’adaptation de contextes. D’autre part, il traite, de manière optimale, du nombre et du placement des capteurs/effecteurs avec la programmation par contrainte.

Objectifs

Le but de la thèse est de présenter un modèle d’organisation de la connaissance ambiante dans des habitats intelligents. Ce modèle propose aux personnes ayant peu ou pas de connaissances en Internet des objets de concevoir une solution d’assistance à la réalisation de leurs AVQ ou celles de leur proche. Plus spécifiquement, ce travail se décline en cinq objectifs :

1. Faciliter la mise en place de composants utiles pour offrir aux utilisateurs un écosystème de création de solutions innovantes, tout en limitant le recours à des manipulations au niveau de la programmation et de la configuration ;
2. Offrir une démarche permettant de spécifier et de concevoir un scénario de réalisation d’une AVQ ;
3. Permettre une génération automatique du nombre de capteurs/effecteurs et de leur configuration nécessaires à la réalisation d’un, de plusieurs ou d’un sous-ensemble de scénarios ;
4. Proposer un modèle d’assignation pour le positionnement dans l’habitat intelligent des capteurs/effecteurs/contrôleurs impliqués dans les scénarios ;
5. Offrir des outils basés sur le modèle sémantique ontologique permettant de rendre sensible au contexte l’application d’assistance ambiante.

INTRODUCTION

Le but est d'offrir à l'utilisateur la possibilité de créer de la valeur en combinant plusieurs aspects plutôt que de restreindre ses choix à un ensemble de paramètres fonctionnels ou système. Ce processus se fait par la compréhension de la démarche de l'utilisateur via la création et l'exécution des scénarios. Nous visons donc à mettre en place des outils proches de certains diagrammes UML (Unified Modeling Language) /Entité Relation (ER), mais ayant une syntaxe formelle et une sémantique bien définie. Chacun des modèles produits devrait fournir des contraintes précises pour faciliter le positionnement des dispositifs dans l'habitat. L'ensemble devra être orchestré par une API (Application Programming Interface) utilisant les techniques de la logique de description pour fournir un raisonnement de haut niveau aux exigences de l'utilisateur.

Hypothèses

Nous voulons ici indiquer quelles sont les hypothèses que nous nous sommes fixées pour apporter une solution à notre questionnement. Toutes choses étant égales par ailleurs, les systèmes ambiants DIY (Do-It-Yourself) devraient être aussi efficaces que ceux réalisés par des professionnels.

Première hypothèse : l'innovation se trouve dans l'assemblage. Ceux qui ont inventé les téléphones intelligents n'ont pas inventé les écrans tactiles. On peut dire la même chose des avions et des moteurs à réaction ; plusieurs faits permettent de valider cette hypothèse. Le concept DIY s'inspire de cette réalité. En effet, la connaissance et la technique se construisent par bloc et par assemblage de composants. Une approche DIY permet à tout le monde d'être le constructeur d'une telle connaissance. Il nous revient donc de mettre en place des mesures nécessaires pour que des personnes soient capables de donner le meilleur d'eux-mêmes. Nous pensons donc que la ré modélisation d'outils pour habitat intelligent adossé à l'informatique diffuse offrira un support permettant à chaque personne d'innover dans la construction de son modèle d'assistance.

INTRODUCTION

Deuxième hypothèse : la différence est dans le détail. O combien une situation ambiguë peut causer des dégâts inimaginables et inestimables ! Une situation peut avoir plusieurs interprétations, une assistance donnée peut fonctionner différemment selon plusieurs utilisateurs. Notre hypothèse est donc d’associer à chaque instance d’activités des informations contextuelles permettant de lever toute ambiguïté à la situation. Les informations composant le contexte ne sont pas identiques et se modifient lorsque la situation change. Il faut promouvoir une sémantique commune même si elle n’est pas complète ; une sémantique unique même si elle est insuffisante ; une sémantique ouverte plutôt que fermée.

Troisième hypothèse : séparation des pouvoirs. Pour rendre le système adaptable au profil, il est important qu’il soit sensible au contexte de la personne. Il faut ainsi qu’une séparation (indépendante) existe entre la donnée collectée et les applications utilisatrices. Ceci nous oblige donc à développer les applications sous forme de modules, de briques et de couches qui s’échangeront des données entre elles pour arriver à un consensus sans qu’elles aient une connaissance globale de tous les services offerts.

Quatrième hypothèse : l’union fait la force & l’intelligence est collective. À l’instar des logiciels libres, nous espérons avoir des kits DIY libres. L’hypothèse de base est de penser que si une communauté se développe autour du concept DIY, alors les kits déjà construits peuvent apprendre et collaborer entre eux pour offrir plus de services. Nous souhaitons prendre en compte plusieurs scénarios individuels afin d’offrir une diversité et une richesse multidisciplinaire du concept d’assistance. À cet effet, les infrastructures individuelles deviendront des supports au développement collectif, à des intelligences collectives et citoyennes.

Méthodologie

Pour soutenir notre argumentation et mener à bien nos travaux, nous avons opté pour une démarche itérative comportant plusieurs étapes de validation. Pour faire suite à la preuve de concept que nous avons réalisé au sein du laboratoire DOMUS sur

INTRODUCTION

l'errance nocturne [88, 89], une recherche bibliographique a été menée afin d'identifier les travaux et les solutions utilisées pour offrir une telle assistance. L'objectif principal de cette revue était de nous positionner afin d'identifier clairement les concepts théoriques et les méthodologies scientifiques validés et utilisés dans le domaine de l'assistance ambiante. Puis de manière théorique, nous les avons comparés à notre processus pour dégager celles qui produiraient des exceptions dans notre preuve de concept. Ce premier jalon a donné lieu à un premier niveau de questionnement. En effet, très peu de solutions utilisaient une approche centrée utilisateur et parmi elles un très faible nombre offraient la possibilité ou la capacité aux utilisateurs n'ayant pas ou peu de connaissances de construire eux-mêmes leurs assistances.

Parallèlement, nous nous sommes intéressés aux solutions qui offraient la capacité de prendre en compte les besoins de l'utilisateur tout en étant aux faits des exigences de son profil de santé et des exigences techniques. Les seules qui se dégageaient étaient celles utilisant les ontologies. En même temps que les ontologies permettaient l'expression des besoins à un niveau élevé et assuraient des raisonnements par inférences, elles structuraient la connaissance sur la personne et l'assistance requise. De plus, un précédent travail de maîtrise par Marc Chevalaz au laboratoire DOMUS avait démontré qu'il était possible d'utiliser un dépôt pour la gestion des événements temps réel dans une ontologie [25]. Cependant, très peu de solutions basées sur les ontologies structuraient la totalité de la connaissance ambiante pour permettre aux utilisateurs finaux de concevoir leurs assistances.

L'analyse des résultats de la première expérimentation et les exigences multidisciplinaires du travail sur le terrain ont clairement orienté la démarche du processus vers une conception centrée utilisateur dirigée par les scénarios.

La validation de l'étape de la revue de la littérature nous a permis de constater que la question de la construction de l'assistance ambiante qui prend en compte tous les aspects (besoin, scénarios, utilisateurs, habitat, assistance, capteurs, effecteurs, applications...) n'était pas abordée d'un point de vue du DIY. La méconnaissance des contraintes que l'on peut rencontrer dans cette direction nous a clairement donné envie de l'explorer davantage. Nous voulons par ce fait étendre la connaissance sur l'assistance ambiante pour une communauté DIY.

À chaque cycle, nous avons produit un prototype testé et validé au sein du labo-

INTRODUCTION

ratoire. Le premier prototype de l'application d'assistance d'errance nocturne a servi dans des expérimentations à domicile. Le second prototype concerne la production d'une application pour spécifier de manière formelle les scénarios et les valider dans une approche multidisciplinaire. Le troisième prototype est celui de la construction de l'ontologie pour décrire le contexte d'une application d'assistance ambiante. Le dernier prototype est celui de la production du modèle de satisfaction de contraintes pour le positionnement des capteurs dans un habitat intelligent. Dans des processus de raffinement successifs, chaque prototype a servi de base au suivant, nous permettant ainsi de valider les acquis. Cette démarche s'inscrit dans le prototypage et l'évaluation continus. Observations expérimentales, tests logiciels, analyse du profil utilisateur, conception dirigée par les scénarios furent les outils à notre disposition pour compléter et valider cette étape.

Dans le cadre de la conception de maison intelligente, le modèle sémantique est considéré comme une étape intermédiaire apportant la connaissance contextuelle nécessaire à la réalisation d'un besoin précis. Dès lors, il s'impose comme un outil capable d'utiliser des raisonneurs pour inférer des connaissances à un très haut niveau. La démarche méthodologique proposée s'intègre dans un processus d'ingénierie logicielle, de modélisation de la connaissance par une démarche de conception par prototypes itératifs. Elle s'appuie sur les standards de résolution algorithmique, de vérification formelle, de validation des spécifications, de satisfaction de contraintes et d'architecture logicielles pour produire un modèle offrant les primitives nécessaires à l'expression de la sensibilité au contexte des applications d'assistances ambiantes.

Résultats

Le modèle développé est testé et expérimenté au sein du laboratoire DOMUS. La preuve de concept menée dans le cadre d'une expérimentation à domicile chez une personne âgée souffrant d'une démence de type Alzheimer nous a permis d'affiner les spécifications et de collecter les informations pour enrichir le modèle. La version finale du modèle, comme son API sémantique est utilisée dans un kit d'assistance aux AVQ déployé et utilisé au sein du laboratoire en permanence. Plusieurs publications scientifiques accompagnent et valident les travaux menés dans le cadre de cette étude

INTRODUCTION

[89, 88, 65, 80, 86, 58, 52, 18]. Nous envisageons de faire des simulations à grande échelle pour tester certaines généralités de notre modèle.

Structure de la thèse

Organisée en huit chapitres, cette thèse aborde dans son Chapitre 1 les techniques et outils utilisés dans la littérature pour l’intelligence ambiante. Il commence par mettre en avant-plan le caractère ubiquitaire de l’informatique, il se base sur cet aspect diffus de l’informatique pour présenter la question de la compréhension de l’assistance ambiante et de l’autonomie aux personnes âgées. Un rapide tour des connaissances et des définitions sur le contexte et comment les applications peuvent y être sensibles est présenté. Un état des solutions se basant sur les ontologies pour construire des connaissances sur l’assistance est abordé. Le chapitre continue en abordant la relation qui existe entre l’Internet des objets, l’informatique diffuse et la sensibilité au contexte d’un point de vue de l’intelligence ambiante. Il se termine en présentant les aspects liés au paradigme DIY.

Le Chapitre 2 présente quelques solutions technologiques mettant en place des mécanismes d’habitats intelligents. Il fait une revue de la littérature des projets de recherches de laboratoire d’assistance aux AVQ, avec un accent particulier sur ceux permettant de faciliter le maintien à domicile ou la reprise d’autonomie aux personnes âgées. Cette partie conclut l’état de l’art de cette thèse.

Le Chapitre 3 introduit nos contributions en présentant une approche de spécification et de validation des scénarios utilisées pour construire une assistance aux AVQ. Il commence par une définition des termes clés nécessaires à l’implantation d’une approche DIY. Il décrit le processus centré utilisateur de construction des scénarios. Ce chapitre montre comment dans une approche multidisciplinaire, l’utilisation d’un outil (ALLOY) de spécifications formelles permet de spécifier, de valider et d’assurer la conformité du processus hiérarchique de création de scénarios. Il présente différentes méthodes inspirées du DIY pour construire des scénarios. Une attention particulière est mise sur le scénario d’errance nocturne.

Le Chapitre 4 présente le modèle ontologique développé dans le cadre du projet AGE-WELL. Il commence par mettre en évidence la nécessité d’avoir un modèle

INTRODUCTION

sémantique. Il décrit par la suite, tour à tour la construction de l'ontologie d'un point de vue de la modélisation de l'environnement, d'un point de vue de la modélisation de scénarios et d'un point de vue de la modélisation de dispositifs électroniques (capteurs, effecteurs, contrôleurs...).

Le Chapitre 5 présente les détails d'implémentation de l'API sémantique nécessaire pour communiquer avec l'ontologie. Il présente l'architecture fonctionnelle et logicielle puis les technologies et les outils de conception utilisés pour bâtir l'API.

Le Chapitre 6 est consacré à la définition et à la mise en place d'une architecture logicielle pour prendre en compte des mécanismes de sensibilité au contexte. L'architecture proposée se présente comme la dorsale pour la définition des applications DIY.

Le Chapitre 7 aborde le modèle de programmation linéaire mis en place pour approcher la question du positionnement des capteurs dans l'environnement. Il décrit comment ce modèle d'optimisation de contraintes se sert du contexte mis en place et est maintenu par le modèle ontologique pour déterminer la position éventuelle qu'occuperait un capteur dans l'habitat intelligent.

Le Chapitre 8 aborde les détails et les résultats de l'expérimentation à domicile effectuée et des tests effectués en laboratoire. Il présente aussi une évaluation des objectifs de cette thèse.

Ce document se termine par une conclusion qui revient sur les points importants et les contributions. Elle fait une synthèse sur les travaux futurs et les perspectives de recherche induites et satellitaires à ce travail.

Première partie

État de l'art

Chapitre 1

Intelligence Ambiante, Sensibilité au Contexte, DIY & Ontologie

L'intelligence ambiante comme un chef d'orchestre apporte au sein d'un habitat une symbiose, une adaptation et une synchronisation entre l'environnement, l'utilisateur, le profil utilisateur et les dispositifs électroniques de l'Internet des objets. Cette harmonie est facilitée par l'apport de l'intelligence artificielle comme support à la coordination des composantes de l'environnement. Trois grandes disciplines s'organisent pour former l'intelligence ambiante : l'informatique diffuse, la sensibilité au contexte et l'intelligence artificielle.

Ce chapitre précise et définit tous ces termes. Il présente dans sa section 1 les travaux qui ont été réalisés dans le domaine de l'informatique ubiquitaire. La section 2 utilise les avancées de l'informatique ubiquitaire pour définir et présenter le contexte et les solutions de sensibilité au contexte. La section 3 marque un point d'arrêt sur les ontologies et leur apport pour l'intelligence ambiante. La section 4 montre comment les concepts abordés plus haut sont mis ensemble pour créer de l'intelligence ambiante. Par la suite, elle présente quelques solutions d'intelligence ambiante disponibles. La section 5 présente les travaux réalisés pour permettre aux solutions d'intelligence ambiante d'être DIY. Nous concluons ce chapitre en présentant les défis qui restent à relever pour une intelligence ambiante utilisant une approche DIY.

1.1 L'informatique diffuse

L'informatique diffuse ou ubiquitaire est une extension de l'informatique classique. Elle apporte une vision de l'informatique moderne où, les agents logiciels, les services et les dispositifs électroniques agissent, s'intègrent et coopèrent de manière calme et transparente pour soutenir les objectifs des personnes. Quand nous parlons de calme, il s'agit de technologies qui produisent et autorisent des interactions à la périphérie de l'utilisateur plutôt qu'au centre de son attention. Les objectifs des personnes peuvent être de diverses sortes, du plus simple au plus complexe. Concrètement, le but de l'informatique diffuse est d'offrir aux applications l'utilisant les capacités d'anticiper sur nos besoins, de négocier et faire la médiation de services, d'agir en notre nom, d'offrir des services continus en tout lieu et à tout moment, etc.

1.1.1 Contexte et définition

L'informatique diffuse, pervasive, ou ubiquitaire offre (conçoit) une nouvelle façon d'utiliser l'informatique, qui prend en compte l'environnement humain, met l'ordinateur en arrière-plan et l'humain au centre de l'attention [116]. Plus simplement, il s'agit de la capacité de multiplier et de fortement décentraliser les unités de calcul et des dispositifs électroniques de communication autour de l'humain. Cette croissance est due à la forte évolution technologique et au bas coût des capteurs, des effecteurs, des microprocesseurs, et des équipements de communication sans fil. Les environnements ubiquitaires se caractérisent donc par :

1. La transparence/invisibilité du dispositif électronique installé. Il s'agit ici de considérer tous les dispositifs électroniques comme étant des domestiques tranquilles et invisibles.
2. La répartition des services entre équipements mobiles et fixes. Les dispositifs ont pour but de nous aider à accomplir un besoin précis. L'intelligence induite vient de la manière intuitive dont on réalise des choses. Ces dispositifs sont considérés comme étant une extension de notre inconscient.
3. L'hétérogénéité des équipements et la technologie doivent créer le calme. La multiplication des dispositifs et des services ne devrait pas nuire, mais travailler

1.1. L'INFORMATIQUE DIFFUSE

ensemble dans un accord presque parfait pour valoriser les utilisateurs.

1.1.2 Informatique ubiquitaire – Technologie calme et DIY

Weiser, en 1991, présentait une vision d'une informatique non intrusive et invisible qui accompagnerait les utilisateurs à la réalisation de leur objectif. Il parlait de l'informatique du 21^e siècle [115]. Malheureusement, cette vision ne s'est pas démocratisée partout et son impact dans les environnements ubiquitaires suit des directions parfois satellitaires à la vision initiale. En 1997, Weiser propose une nouvelle version intitulée l'arrivée de l'âge des technologies calmes [117]. Dans ce document, il met en avant le fait que ce qui compte c'est la relation que les gens entretiennent avec la technologie. Plusieurs travaux ont suivi ce concept et ont donné naissance aux principes de technologies calmes résumés en 8 points (voir annexe A) [20]. Le but de la vision de Weiser est de rechercher une technologie exigeant le minimum d'attention possible pour satisfaire les besoins des personnes. La mise en place et la réalisation des AVQ varient généralement d'une personne à une autre et sont parfois modélisées en utilisant des technologies simplifiant à un niveau basique leur complexité. Le but étant de donner plus de pouvoir aux concepteurs. De ce fait, cette simplification rend l'utilisation des solutions proposées dans la pratique complexes et mal adaptées au besoin initial qui du coup s'oppose à la vision initiale de Mark Weiser. Pour comprendre et s'interroger sur le retard ou la faible implantation de la vision proposée par Weiser, il peut être intéressant de donner la parole ou le pouvoir aux personnes (utilisateurs).

Dans son article, Yvonne Rogers [94] introduit un nouveau paradigme. Elle constate simplement que les solutions apportées ne permettent toujours pas de satisfaire la vision de Weiser. Son nouveau paradigme est axé sur l'engagement plutôt que sur l'apaisement des personnes. Dans son approche, elle estime que les humains ont plus de facilité pour exploiter leurs environnements et étendre leurs capacités en utilisant les stratégies et les outils existants. Elle préconise donc une approche où les gens ont plus de potentiel pour combiner, adapter et utiliser les technologies de manière créative et constructive. Sachant que généralement les gens ne désirent pas dépendre des ordinateurs dans la réalisation de leurs tâches, mais tiennent pour la plupart, à être actifs et garder le contrôle. Il devient donc important de ratisser plus large et

1.2. LA MODÉLISATION DU CONTEXTE

d'ouvrir les applications à l'engagement de la créativité à l'innovation et à l'imagination des gens. L'informatique ubiquitaire favorise la présence dans l'environnement de multiples dispositifs transparents et interconnectés, mais ne définit pas comment ils devraient fonctionner entre eux. La sensibilité au contexte permet d'apporter des solutions à ces manquements.

1.2 La modélisation du contexte

1.2.1 Le contexte

Dans la vie de chaque jour, nous utilisons le contexte pour prendre des décisions. Il est parfois le centre de notre conversation et se manifeste de manière implicite ou explicite. Les données de contexte se révèlent donc capitales pour la compréhension de la situation dans laquelle on se trouve.

Une mauvaise compréhension du contexte peut entraîner de lourdes conséquences. Considérons le scénario d'assistance à domicile pour une personne âgée atteinte de la maladie d'Alzheimer. Si elle entre dans une pièce, alors allumer la lumière dans cette pièce, puis quand il n'y a plus de mouvements dans la pièce éteindre cette lumière. Il est clair que ce scénario ne fonctionne pas, car plusieurs informations contextuelles ne sont pas prises en compte durant l'exécution du scénario. Nous savons que les personnes âgées font très peu de mouvement et sont parfois à mobilité réduite donc elles peuvent regarder la télévision, s'allonger sur le divan ou rester immobiles pendant des minutes sans rien faire. On voit ici qu'aucune information sur le quoi et le comment de la personne, n'a été prise en compte et au risque d'éteindre la lumière à mauvais escient.

Comme le dit A. Dey [34], l'utilisation du contexte est importante dans les applications interactives et plus importantes encore pour les applications dans lesquelles le contexte de l'utilisateur change rapidement. La définition apportée par Schilit et al [97] pour la notion de contexte met en avant trois aspects importants : **où est la personne ; qui est avec la personne ; et quelles ressources sont proches de la personne ?** Cette définition implique une surveillance constante de l'environnement dans lequel évolue la personne avec une approche centrée sur la personne.

1.2. LA MODÉLISATION DU CONTEXTE

Une autre définition proche de celle-ci est celle donnée par Pascoe [81] qui pense que le contexte joue un rôle important dans le développement des capteurs portables. À cause du caractère dynamique de l’environnement d’exécution et de la nature fortement ad hoc du système ubiquitaire, il est impossible d’énumérer quels aspects d’une situation sont importants. Cela a comme incidence le fait que d’une situation à une autre certains aspects peuvent devenir moins pertinents, voire inutiles dans l’interaction.

La définition qui revient le plus dans la littérature concernant le contexte est celle donnée par Dey et al [33] en 2001. « Le contexte est toute information que l’on peut utiliser pour caractériser la situation d’une entité. Une entité est une personne, une place, ou un objet considéré comme important pour l’interaction entre l’utilisateur et l’application, incluant l’utilisateur et l’application elle-même ». Elle implique que si un élément d’information peut être utilisé pour caractériser la situation d’une personne dans une interaction, alors cette information est le contexte. Ainsi les concepts mis en avant par Schilit ont été augmentés, on retrouve maintenant la localisation « *where* », l’identification « *who* », l’activité « *what* » et le temps « *when* ». L’ajout de ces concepts permet d’enrichir le contexte avec de nouvelles informations.

Alors que la définition de Schilit et al [97] ne proposait que la localisation et l’identification pour comprendre le contexte sans inclure ce que fait l’utilisateur. Nous allons retenir dans ce document celle proposée par Dey et al [33].

1.2.2 La modélisation du contexte

La gestion du contexte est vitale pour la mise en place d’une solution utilisant l’informatique diffuse. Elle permet aux développeurs et aux architectes de services de fournir aux utilisateurs finaux une simplicité d’utilisation, une action et une réaction dynamique du système, une adaptation au changement et au contenu des données. Le contexte constitue donc un élément important et essentiel du comportement du service, surtout quand les interactions avec les utilisateurs finaux sont au cœur du système. Dans la littérature, il existe cinq grandes catégories de modélisation de contexte [64] :

1. Les solutions basées sur les intergiciels et les plateformes de services dédiées ;

1.2. LA MODÉLISATION DU CONTEXTE

2. L'utilisation des ontologies ;
3. L'utilisation des raisonnements basés sur les règles logiques ;
4. L'utilisation des extensions au niveau de la programmation ;
5. Les approches basées sur des modèles ;

Il est important de déterminer comment le système peut avoir connaissance de son contexte et d'être capable de s'adapter à celui-ci. Les outils de modélisation proposés permettent de rendre les applications sensibles au contexte.

1.2.3 La sensibilité du contexte

Comment le système peut-il être conscient de son environnement ? Comment s'adapte-t-il aux changements ? Comment interagit-il avec son environnement ? Telles sont les questions auxquelles cette section apportera des réponses.

La sensibilité au contexte (*Context awareness*) a été introduite en 1994 par Schilit et al [97] comme étant le fait qu'une application s'adapte à son environnement. Cette adaptation est fonction de son lieu d'utilisation, de l'ensemble d'objets impliqués, des personnes à proximité et des modifications apportées à ces objets au fil du temps. On passe donc d'une application qui permet d'informer sur le contexte à une application capable de s'adapter à son contexte. Il devient donc plus commun de rapprocher le terme de ses synonymes : adaptatif, réactif, sensible, situation, environnement direct [33].

Plus formellement, un système est sensible au contexte [2, 34, 33] s'il utilise le contexte pour fournir des informations ou des services pertinents aux utilisateurs. La pertinence dépend de la tâche de l'utilisateur. De cette définition, on peut voir que plusieurs opérations sont nécessaires pour la mise en place d'un système sensible au contexte. :

1. La collecte d'informations de contexte ;
2. La détection des changements de contexte ;
3. La réalisation des adaptations au contexte.

Ces actions peuvent être menées sans intervention humaine. Par conséquent, il est important que les systèmes informatiques réagissent au changement de contexte, peu

1.2. LA MODÉLISATION DU CONTEXTE

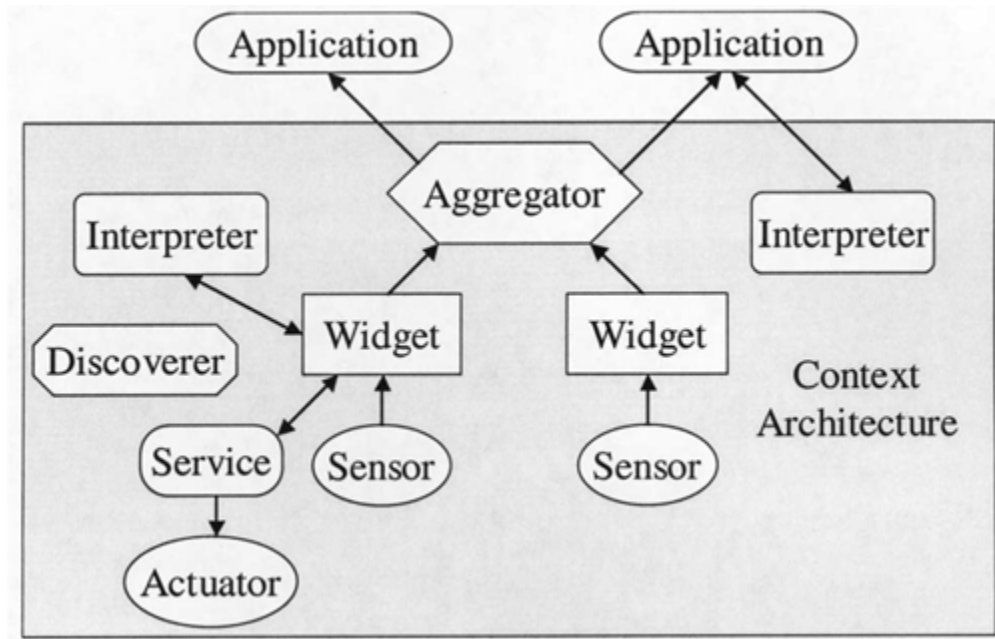


Figure 1.1 – Architecture du Context Toolkit components [33]

importe l'initiateur.

Plusieurs architectures génériques de développement d'applications sensibles au contexte offrent des services divers à plusieurs niveaux. Le modèle proposé par Dey intitulé la boîte à outils de contexte «*Context Toolkit*» [33] est un cadriceil permettant de développer des applications sensibles au contexte. Ce cadriceil est composé d'outils découplés de la partie architecture contextuelle de l'application et des problématiques métiers. Dey préconise une organisation à plat où les capteurs sont directement connectés à un module de *widgets*. Des interpréteurs sont définis pour le calcul et l'inférence de contextes interprétés. Les agrégateurs permettent de fusionner les données des entités observables. Un agent est défini pour assurer la gestion des opérations d'adhésion. Le cadre est construit de manière à permettre que plusieurs *toolkits* distincts puissent interagir. La Figure 1.1 montre l'entrelacement des modules pour rendre le système adaptable au contexte. Contrairement à la première solution, on a un découplage entre applications et gestion de contexte.

1.3 Ontologie

Nous nous intéressons dans cette partie à l'utilisation des ontologies comme solution à la sensibilité au contexte. Le formalisme sous-jacent aux ontologies permet de réduire l'espace de recherche des concepts en éliminant de nombreuses expressions sans signification, et contradictoires. De plus, les ontologies permettent de codifier les connaissances en les rendant réutilisables par des personnes ou des machines. La mise en place d'une sémantique bien définie et la disponibilité des outils de raisonnement permettent d'avoir des ontologies de haute qualité. Cette section avant, d'aller plus loin, traite de la contextualisation des données, de leur traitement et de leur représentation.

1.3.1 Données – Informations – Connaissances

Pour générer de la connaissance, il est important de formaliser la notion de donnée. Le passage d'un ensemble de données à une connaissance est réalisé à travers une abstraction intermédiaire qui est l'information. Bien que ces trois termes soient utilisés de manière abusive pour désigner la même entité, nous allons dans cette section apporter quelques éclaircissements à la compréhension de chaque terme et aux relations existantes entre eux.

La donnée

Mesure brute typée ou non, la donnée est une valeur obtenue d'un dispositif électronique. Elle représente donc un fait, une observation quantifiable. Généralement collectée, stockée, traitée, transmise et analysée, la donnée n'a aucune signification prise individuellement. Pour être pertinente, elle doit être associée à un contexte bien précis. Dans ce cas, on parle alors d'information.

L'information

L'information offre à la donnée son contexte d'analyse et de traitement. L'information devient alors compréhensible à la fois par les humains et par les machines et donne un sens particulier à chaque donnée. Porteuse de sens, l'information permet de

1.3. ONTOLOGIE

mieux répondre aux questions Quand ? Où ? Qui ? Quoi ? etc. La réponse apportée à ces simples questions ne permet pas de faire des déductions. Pour y parvenir, il faut monter en abstraction vers la connaissance.

La connaissance

Formelle et représentable, la connaissance permet d'exprimer (représenter, structurer et organiser) l'information véhiculée par une donnée. Mis ensemble les connaissances permettent de produire de nouvelles connaissances dans un processus d'inférence ou de déduction.

Exemple : de la donnée à la connaissance

Un exemple d'information utilisée dans un habitat intelligent est la valeur de l'état d'un capteur de mouvement d'une chambre à coucher. La valeur de la donnée "ON" n'a de sens que si on lui associe le dispositif et le lieu dans lequel il a été capté. Cette information est ensuite représentée puis encapsulée avec d'autres informations comme l'heure et fusionnée pour donner naissance à une connaissance. Exemple : Le capteur de mouvement de la chambre au-dessus du lit donne "ON" à 22 heures 25 minutes. Cette connaissance est associée à une autre connaissance : tous les soirs à 20 h la personne se couche sur son lit pour s'endormir. L'inférence nous permet sur la base de ces connaissances de dire que la personne est sur son lit probablement en train de dormir.

Pour permettre la spécification des connaissances, leur expression, leur représentation et leur partage entre machine ou humain et toutes autres entités, le W3C (World Wide Web Consortium) a mis en place un formalisme orienté graphe basé sur des standards XML.

1.3.2 Les ontologies

Le formalisme mis en place par le W3C possède deux fondements importants, à savoir l'identification et l'adressage de toutes les ressources via les URI (Uniform Resource Identifier). La représentation des liens entre les entités par une relation ternaire est basée sur les triplets RDF (Resource Description Framework). Un triplet

1.3. ONTOLOGIE

RDF est une association (prédicat, sujet, objet). Le sujet représente la ressource à décrire, le prédicat représente un type de propriété applicable à cette ressource, l'objet représente une donnée ou une autre ressource : c'est la valeur de la propriété.

Les URI sont une représentation sous forme de chaîne de caractères permettant d'identifier de manière unique une entité (objet, attributs, texte, valeur, etc.). Le Format RDF utilise trois composantes principales pour représenter une connaissance : le sujet, le prédicat et l'objet. Très vite, le vocabulaire RDF des triplets qui propose une définition hiérarchique entre les entités a montré ses limites. Par la suite, il a été associé à un nouveau vocabulaire beaucoup plus riche et plus formel, le OWL (Ontology Web Language). OWL est une spécification permettant d'exprimer de manière verticale et horizontale des relations existantes entre les entités.

Plusieurs définitions des ontologies dans la littérature utilisent le langage OWL pour structurer les connaissances. Parmi les plus utilisées figure celle de Gruber [46, 47] qui définit une ontologie comme une spécification explicite d'une conceptualisation. Les termes "spécification explicite" et "conceptualisation" sont ici d'une importance majeure. Nous y reviendrons. Pour compléter, Noy [78] définit une ontologie comme étant la représentation formelle, explicite, identifiable et consensuelle de l'ensemble des concepts partagés d'un domaine sous forme de classes, de propriétés et de relations qui les lient. Nous notons dans cette définition la présence des termes clés identifiés plus haut et l'apparition de nouveaux termes qui insistent sur les éléments permettant de matérialiser une ontologie. Dans ce document, nous retiendrons la définition émise par Studer et al [108] dans l'ingénierie des connaissances. Elle définit l'ontologie par une spécification formelle explicite d'une conceptualisation partagée d'un domaine donné. Pour mieux comprendre cette définition, Studer [108] apporte un éclaircissement aux termes clés de la définition.

- Formelle indique que la spécification produite doit être lisible sans ambiguïté par une machine ;
- Explicite, signifie une spécification bien formée, c'est-à-dire que les types, les concepts et les contraintes sont explicitement définis ;
- Référençable indique que toutes les entités sont accessibles via un identifiant unique à partir de n'importe quel contexte ;

1.3. ONTOLOGIE

— L’aspect consensuel signifie qu’elle est admise de tous ou par une sous-communauté.

La conceptualisation est la modélisation abstraite du monde réel pour la détermination des classes, des entités, des attributs, des relations et des axiomes qui le mettent en place. Le concept de partage est lié au consensus, à la généralité et à la réutilisation des connaissances. Quatre types d’ontologies sont définis [108].

1. Les ontologies de domaine, il s’agit d’ontologie qui capture les connaissances d’un domaine particulier (Électronique, Médecine, Mécanique, Alimentaire, etc.). Elle fournit les concepts et les relations permettant de couvrir les vocabulaires, activités et théories de ces domaines ;
2. Les ontologies génériques représentent celles qui sont valides et transversales à plusieurs domaines. En exemple, on peut citer les ontologies sur les personnes, les mesures, etc. Ces ontologies sont souvent aussi désignées par ontologie cœur ou ontologie de haut niveau ;
3. Les ontologies d’applications contiennent quant à elles toutes les connaissances très spécifiques à la modélisation d’un domaine particulier. Contrairement à l’ontologie du domaine, elle est plus spécifique et non réutilisable.
4. Les ontologies représentationnelles ne modélisent aucun domaine en particulier, mais des entités de représentation sans indiquer ce qui devrait être représenté. Exemple l’ontologie « Frame Ontology » qui définit les concepts tels que les cadres, les créneaux horaires et les contraintes des créneaux permettant d’exprimer des connaissances dans un langage orienté cadre.

1.3.3 Langage et formalisme

Les formalismes ontologiques mis en place pour exprimer de manière explicite les entités et leurs relations visent à développer un mécanisme de gestion et d’échange de connaissances. Le but est de définir une sémantique des concepts de manière unique via l’utilisation de langages et de formats de représentation des données. Parmi ses langages, nous pouvons citer RDF [68], RDFS [7], Daml, OIL, Daml+OIL [57] et OWL [91, 74]. Le W3C considère OWL comme étant le standard le plus abouti pour la définition et la représentation des ontologies. Cependant, il existe trois familles

1.4. SPÉCIFICATIONS FORMELLES

affiliées à OWL. OWL Lite est une version allégée de OWL. OWL-DL est défini comme étant une extension d'OWL Lite. Et OWL Full est une extension de OWL DL. Ces variétés de langage OWL permettent simplement d'établir un compromis entre le raisonneur et la représentation des données. En effet, plus un langage est expressif et libre syntaxiquement d'un point de vue OWL mieux il offre de bonnes capacités de raisonnement et une grande complexité de calculabilité. Ainsi, une famille de OWL sera choisie en fonction des besoins.

Avant de voir comment les ontologies sont utilisées pour construire le contexte et offrir une assistance ambiante, nous allons voir pourquoi et comment il est nécessaire de spécifier

1.4 Spécifications formelles

Quelle démarche employer pour s'assurer que l'expression des besoins par les scénarios est conforme au contexte donné ? Comment s'assurer que ces besoins n'entrent pas en contradiction ? Peut-on décrire et prédire toutes les situations pouvant se produire dans un scénario ? Si tel n'est pas le cas alors comment pourrait-on garantir le fait qu'une nouvelle situation non connue restera conforme à l'esprit initial ?

Une spécification est la description précise des éléments constituant une entité. Lorsque celle-ci est écrite dans un langage formel, on dit que c'est une spécification formelle. Une spécification est formelle si elle est fondée sur des modèles et procédés mathématiques, c'est-à-dire qu'elle a une syntaxe formelle et une sémantique formelle et bien définie [61, 14]. À la fin du processus la spécification permet de définir le quoi, mais pas le comment.

1.4.1 Spécification et assistance aux personnes âgées

Le caractère sensible de l'assistance aux personnes âgées via les technologies d'assistance et l'Internet des objets impose un choix judicieux des actions à entreprendre d'une part par le système et d'autre part par l'utilisateur. Le choix des interactions impose d'éliminer au maximum les imprécisions dues aux méthodes de conception logicielles. Le choix des spécifications formelles permet d'avoir à la fois une syntaxe

1.4. SPÉCIFICATIONS FORMELLES

et une sémantique bien définie permettant ainsi de comprendre ce que chaque entité apporte, évitant ainsi d'user d'extensions précises qui rajouteraient une sur-couche de sémantique formelle aux méthodes de modélisation utilisant les diagrammes comme l'UML. Il impose surtout un processus rigoureux, itératif et permettant d'assurer la conformité du système durant la conception, permettant d'énoncer toutes les propriétés sur l'assistance avant même son implémentation. Puisque le but de la spécification est de définir le quoi et que ce pronom relatif est l'élément important pour la prise en compte du contexte, nous pensons qu'il faut spécifier les scénarios d'assistances. Avant de montrer comment nous l'avons mise en œuvre, nous allons présenter une petite revue des outils de modélisation formelle.

1.4.2 Sémantique - Vérification - Raffinement

En utilisant la sémantique et la vérification, les méthodes formelles permettent de décrire des systèmes sans ambiguïté. L'intérêt de la sémantique formelle est de permettre un raisonnement sur les modèles mathématiques. Chaque propriété du système spécifié est soumise à un processus strict de vérification afin de s'assurer qu'il est conforme aux spécifications formulées par la sémantique. La sémantique, qu'elle soit opérationnelle ou dénotationnelle permet une interprétation stricte des symboles de la syntaxe utilisée. Généralement, le processus de vérification se fait soit par preuve ou par vérification de modèle (model-checking).

La preuve [14] utilise des axiomes et des règles d'inférence pour prouver un ensemble de propriétés. Cette vérification qui se fait manuellement ou automatiquement peut nécessiter une intervention humaine lorsque certaines inférences ne sont plus possibles. Généralement, elle est utilisée pour son universalité dans la considération de tous les cas possibles. Elle donne ainsi l'assurance que si le modèle est valide et que la preuve est correcte alors on est assuré du fonctionnement correct du système.

Dans le cas de la vérification de modèle [14, 118, 61], les propriétés sont exprimées sous la forme de logique temporelle, celles-ci sont toutes parcourues afin de trouver un contre-exemple qui violerait une propriété exprimée. Généralement, l'espace d'état du modèle spécifié est fini.

Pour assurer la conformité aux spécifications, on utilise la technique du raffine-

1.4. SPÉCIFICATIONS FORMELLES

ment [118, 72]. Cette technique permet de construire progressivement et de manière itérative des implémentations qui correspondent à la spécification de base. Elle procède étape par étape à des dérivations successives des spécifications données pour s'assurer que chacune d'elles préserve la conformité de base.

1.4.3 Les outils de spécifications

Les méthodes de spécification formelles sont généralement organisées en deux groupes, il s'agit des approches basées sur les états et des approches basées sur les événements.

Les approches basées sur les événements

Elles modélisent la représentation du système par des entités indépendantes appelées agents ou processus. Le comportement du système est décrit par des arbres d'événements ou des séquences d'événements. Parmi les outils disponibles, on retrouve : les réseaux de Petri, Lotos et CSP.

- Les réseaux de Petri utilisent des variables discrètes pour modéliser grâce à des outils graphiques et mathématiques le comportement dynamique des systèmes basés sur des événements [84].
- Le langage LOTOS est un langage algébrique qui utilise la temporalité des événements pour modéliser le système [17].
- CSP (Communicating Sequential Processes) est une algèbre de processus qui intègre des mécanismes de synchronisation sous forme de rendez-vous pour la modélisation des interactions de systèmes [55].

Les approches basées sur les états

Ils ont pour objectif de décrire le système à travers une modélisation statique et dynamique. La modélisation statique permet d'exprimer et de représenter toutes les entités du système et l'état de celui-ci. La modélisation dynamique décrit toute la dynamique du système c'est-à-dire les changements d'état, les interactions entre les entités et les opérations internes et externes. Quelques outils permettant de faire ce

1.5. DO-IT-YOURSELF

type de modélisation sont le langage VDM, le langage Z, Le langage Object-Z, la méthode B et B-event et ALLOY.

- Le langage VDM (Vienna Development Method) [63] est un langage orienté modèle. Il possède une structuration en variables d'état et d'opérations pour offrir un raffinement sur les structures de données.
- Le langage Z est lui aussi orienté modèle et se présente sous forme d'une collection de schémas décrivant les éléments du système [104].
- Le langage Object-Z est une extension du Z pour lui apporter une couche orientée objet [100].
- Les méthodes B [3] et B-event [4] utilisent aussi l'approche orientée modèle pour offrir un support automatisé au raffinement et à la combinaison de modularité.
- ALLOY utilise un procédé orienté objet et la logique de premier ordre pour créer des micros modèles pouvant automatiquement être vérifiés conformément à la spécification de base [61].

1.5 Do-it-Yourself

« *I want to do it myself* », « *I made it myself.* », « *If you can't open it, you don't own it.* » autant d'injonctions qui rendent le concept Do-It-Yourself (DIY) attirant et qui motivent les personnes qui le préconisent. À cela pourraient s'ajouter les charges économiques qui sont réduites du fait de la réutilisation de matériel et du non-paiement d'expert. De plus, le fait de travailler durant des heures nous permet d'accumuler une certaine expérience [93]. Finalement, le concept DIY permet de confirmer le caractère créatif des personnes et d'offrir un sentiment de contrôle. Il offre le plaisir aux personnes, de créer, d'innover sur des concepts pensés et construits par eux-mêmes ou par une communauté à laquelle elles s'identifient.

1.5.1 Contexte et définition

Le terme Do-It-Yourself (DIY) désigne un mouvement sociétal dont le but est d'analyser, de concevoir et de bâtir ou de construire des choses par soi-même. Le

1.5. DO-IT-YOURSELF

concept DIY est né quand les gens se sont réunis et ont réalisé qu'une des façons d'avancer était de faire des choses par eux-mêmes. Bien que le concept DIY soit issu d'un monde non numérique, il prend de plus en plus de l'ampleur en raison de l'omniprésence des médias sociaux et d'autres plateformes. L'un des facteurs clés dans la culture DIY est le sentiment d'appartenance à une communauté [32, 31]. Un exemple illustrant ce concept est la possibilité offerte à toute personne n'ayant pas de connaissance en programmation Web de bâtir son site Web ou un blogue facilement grâce aux CMS et autres gestionnaires de contenus.

Les progrès en technologie de l'information et de la communication dépendent fortement de personnes passionnées et de bricoleurs[32, 42]. À cet effet, pour promouvoir leurs idées et faire rapidement émerger des compétences, ils se sont parfois regroupés en communauté partageant les mêmes centres d'intérêt. Dans des centres de foires de technologies et de développement, plusieurs projets innovants sont présentés et sont la plupart du temps les fleurons des startups [42]. La naissance des plateformes de « *Crowdfunding, Crowdsourcing* » permet d'encourager les utilisateurs à devenir des « *makers* » et favorise la naissance des « *online Communities* ». Il existe en ligne plusieurs communautés qui permettent à des personnes de partager des intérêts communs.

Dale Dougherty, fondateur de Maker et Make Magazine, regroupe les bricoleurs (« *makers* ») en trois [38] types selon leur niveau d'intervention et leur expérience :

1. Les « *Zero to maker* » correspondent aux débutants ;
2. Les « *maker to maker* » sont les personnes s'identifiant comme membre d'une communauté ;
3. Les « *makers to market* » sont principalement des personnes voulant atteindre un certain marché.

Certains auteurs ont défini trois groupes de bricoleurs permettant d'accompagner le processus DIY, cela indépendamment du type de bricoleur choisi [42]. Ces catégories sont :

1. Initier l'inspiration : il s'agit de redonner de l'énergie à la croissance personnelle en suscitant l'intérêt des personnes via des espaces de bricolage ;
2. Maintenir la communauté ; mettre en place des mécanismes autonomes se dé-

1.5. DO-IT-YOURSELF

veloppant de manière organique pour faciliter le partage des informations et un soutien non contraignant aux membres ;

3. Support pour la commercialisation ; il s’agit de proposer des concepts entrepreneuriaux qui non seulement inspirent la communauté du bricolage, mais suscitent également l’esprit d’entreprise nécessaire pour faire passer une création du stade de l’idée à celui de la commercialisation.

1.5.2 DIY et environnements intelligents

Très peu d’études sont faites sur le concept DIY dans les environnements intelligents. Pourtant comme le relèvent [69, 93], le concept DIY est une question importante pour la sensibilité au contexte. Pour mieux comprendre cela, ils ont identifié deux axes importants de la sensibilité au contexte. Ces deux axes devraient automatiquement être pris en compte pour que l’intelligence ambiante ou des applications intelligentes soient utilisées afin de construire des comportements intelligents. Le premier axe est la prise en compte du contexte d’un environnement précis dans lequel les personnes évoluent. Partant du fait que le contexte est quelque chose que les gens font et non quelque chose décrivant des paramètres, il n’est pas possible de le décrire pour chaque situation. Par conséquent, le contexte ne saurait être fixe pour une implémentation précise. Le second axe est la perte de contrôle de la personne induite par la sensibilité au contexte. En effet, pour offrir des solutions efficaces à l’autonomie et à l’assistance, l’utilisateur ne dispose que de peu d’incidence sur la boucle d’actions qui lui est imposée par les développeurs de ces applications.

Marc Roelands et al [93] dans leur étude proposent une typologie du type de création DIY imaginable dans l’Internet des objets. La Figure 1.2 illustre les trois catégories et met en évidence la nature des données échangées. La première comprend un grand réseau de capteurs et d’effecteurs interconnectés permettant aux personnes d’incorporer les données collectées dans des applications DIY, on parle de « *Use Thing Data* ». La seconde catégorie concerne les capteurs et les effecteurs connectés à l’Internet des objets, pour gérer et interagir à distance avec des solutions, on parle d’installation DIY « *Connect Thing* ». La dernière catégorie est celle des interfaces tangibles favorisées par les dispositifs DIY électroniques permettant la construction

1.5. DO-IT-YOURSELF

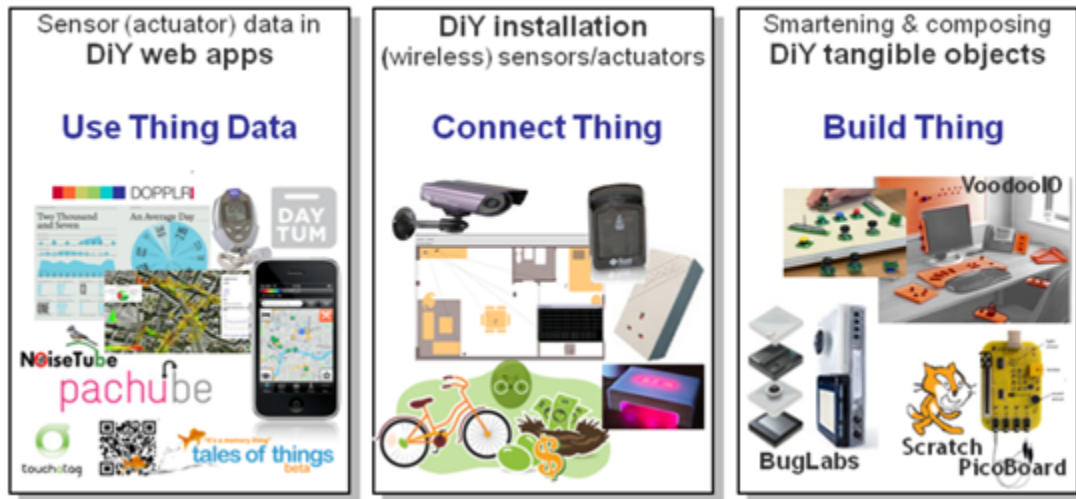


Figure 1.2 – Typologie de création DIY dans l'Internet des objets [93]

des objets connectés, on parle de DIY « *Build Thing* ».

Dans le domaine des habitats intelligents, certains projets ont été conduits dans une philosophie DIY [5, 67, 70, 76, 98, 101, 103, 42, 119]. Pour tester l'efficacité de différents kits DIY sur des scénarios précis de la vie quotidienne, J. Woo et al [119] ont conduit pendant trois semaines une étude dans huit maisons. Cette étude a permis aux auteurs de suggérer un usage des kits DIY pour les maisons intelligentes en six étapes (installation initiale – motivation – implémentation – utilisation à travers plusieurs routines – rodage – suppression) et de relever quelles difficultés les personnes peuvent rencontrer durant l'utilisation d'un tel kit.

Pour sa part, Gary Scott illustre sur un exemple de « *Smart alarm clock* » [98] comment l'utilisation des objets connectés permet de bâtir une communauté DIY. Dans sa thèse intitulée « Programmation d'espace intelligent par l'utilisateur final », Fontaine [40] propose la construction d'espace intelligent sous un angle interface personne-machine par l'utilisateur final. Il s'appuie sur des travaux [101, 73, 103] dans le domaine de la sensibilité au contexte et du concept DIY, pour offrir un environnement sous forme d'interfaces tangibles aux utilisateurs comme support à la traduction en langage informatique des actions à réaliser.

Partant de l'idée que le développement des technologies et de l'Internet des Objets est de plus en plus simple et accessible, il devient évident que la culture numérique

1.6. INTELLIGENCE AMBIANTE

DIY va encourager les gens à la construction de leur propre environnement intelligent. Roeck et al [32, 31] proposent un manifeste en 13 points servant de cadre aux personnes n’ayant pas de connaissances techniques pour la construction de systèmes à base d’outils de l’Internet des objets pour des environnements intelligents. Le manifeste est rédigé sur la base d’expériences conduites durant huit mois avec trente utilisateurs. Pour plus de détails, voir (Annexe B).

À la lecture de ces travaux il en ressort que, les solutions disponibles sont basées sur un paradigme suivant lequel les développeurs sont les « *maker* ». En ce sens, ils programment les commandes qui devraient être exécutées par les kits. Ceux-ci offrent un environnement de programmation visuel basé sur des modèles prédéfinis à des utilisateurs n’ayant pas de connaissances en informatique et qui restent confinés dans cet environnement de commandes. Sur le plan écologique, il offre une meilleure gestion de la consommation d’énergie et permet de respecter les exigences propres à chacun. Sur le plan économique, il offre une meilleure réduction de coûts d’installation et d’exploitation. Sur le plan social et communautaire, on note une meilleure aisance à parler de la solution mise en place et sur le plan moral, une fierté d’avoir accompli quelque chose.

1.6 Intelligence ambiante

C’est à la fin des années 2000 que la miniaturisation des dispositifs électroniques, l’évolution technologique et les bonnes avancées en intelligence artificielle ont permis la naissance de l’AmI (Intelligence Ambiante). L’objectif principal de ce domaine est d’apporter une réponse précise aux besoins de l’utilisateur de manière transparente, calme et non intrusive.

1.6.1 Contexte et définition

L’intelligence ambiante naît du souci de répondre au paradigme proposé par Mark Weiser. En effet, il annonçait que l’on passerait d’un ordinateur par personne à plusieurs ordinateurs par personne comme le montre le schéma de la Figure 1.3.

Plusieurs définitions [29, 28, 30, 13, 41] ont été apportées dans la littérature au

1.6. INTELLIGENCE AMBIANTE

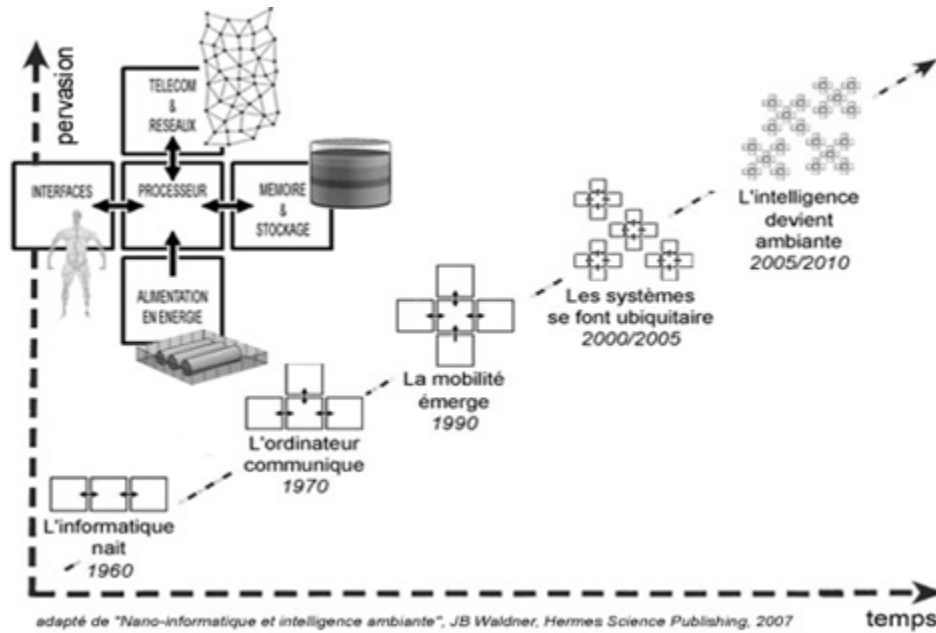


Figure 1.3 – Évolution de l’informatique vers l’intelligence ambiante de Waldner 2007

terme intelligence ambiante. Elles s’accordent plus ou moins sur le fait qu’il désigne la capacité pour un environnement donné, comprenant des personnes et des dispositifs électroniques embarqués, d’être capable de détecter leur état, d’anticiper et de s’adapter éventuellement aux besoins des personnes. Punie [85] définit l’intelligence ambiante comme étant la capacité à se référer à une vision future de la société où les interfaces intelligentes permettraient aux personnes et à des dispositifs d’interagir les uns avec les autres y compris avec l’environnement. Par la suite, il montre le rôle que peut jouer l’intelligence ambiante dans la vie quotidienne. Il termine son étude en mettant l’accent sur les défis et les goulots d’étranglement auxquels est confronté le concept. Les auteurs Aarts et al [1] pensent que la réalisation de l’intelligence ambiante nécessite l’intégration de différents sujets scientifiques, tels que les interactions personne-machine, les interfaces tangibles, les intergiciels et les systèmes multi agents. Ils définissent des mécanismes ambiants comme ceux permettant aux différentes applications de soutenir de façon proactive les utilisateurs, plutôt que de les remplacer. Par des illustrations de plusieurs scénarios d’usage, Ducatel et al [39] font ressortir les besoins en intelligence ambiante. L’intelligence ambiante est décrite comme une

1.6. INTELLIGENCE AMBIANTE

vision de la société où les interactions humaines avec les objets de l'environnement sont facilitées par l'usage de l'intelligence artificielle. Il est facile de constater avec [29, 28, 39], que l'accent est mis sur le fait que les personnes sont entourées par des interfaces intuitives et intelligentes incorporées dans différents objets de la vie quotidienne. L'idée est de permettre à l'environnement de reconnaître et de réagir à la présence de différents individus d'une manière transparente, discrète et souvent invisible. Lorsque les services accompagnent et suivent les utilisateurs dans un environnement ubiquitaire, on dit qu'ils sont ambiants. Pour y parvenir, ils font appel à des contributions d'autres domaines comme l'apprentissage automatique, les systèmes multi agents ou les ontologies.

1.6.2 Les domaines des systèmes d'intelligence ambiante

L'intelligence ambiante représente un domaine bien plus large que la disponibilité transparente des ressources. Elle nécessite une association avec les domaines comme l'intelligence artificielle pour offrir plus d'assistance (intelligente). Son but principal est d'assister l'humain dans ses activités par une coordination intelligente de l'ensemble des appareils mis en communication. Un système d'intelligence ambiante peut-être caractérisé par :

- Un mécanisme d'apprentissage des habitudes, des préférences, et des besoins des utilisateurs.
- Un mécanisme d'adaptation des modèles au cours du temps pour prendre en compte de nouveaux comportements ;
- Un mécanisme de définition du contexte pour faciliter la prise de décision ;
- Un mécanisme pour l'interaction humaine qui repose sur les gestes familiers et naturels sans aucune contrainte supplémentaire ;
- Un environnement ubiquitaire, dans lequel les dispositifs électroniques sont disséminés de manière non intrusive et transparente ;
- Un procédé de suivi et de journalisation des événements ;
- Un module d'intelligence ou une base de connaissances où l'intelligence est caractérisée par la capacité du système à produire une analyse et à mettre en œuvre des actions sur la base de cette analyse.

1.6. INTELLIGENCE AMBIANTE

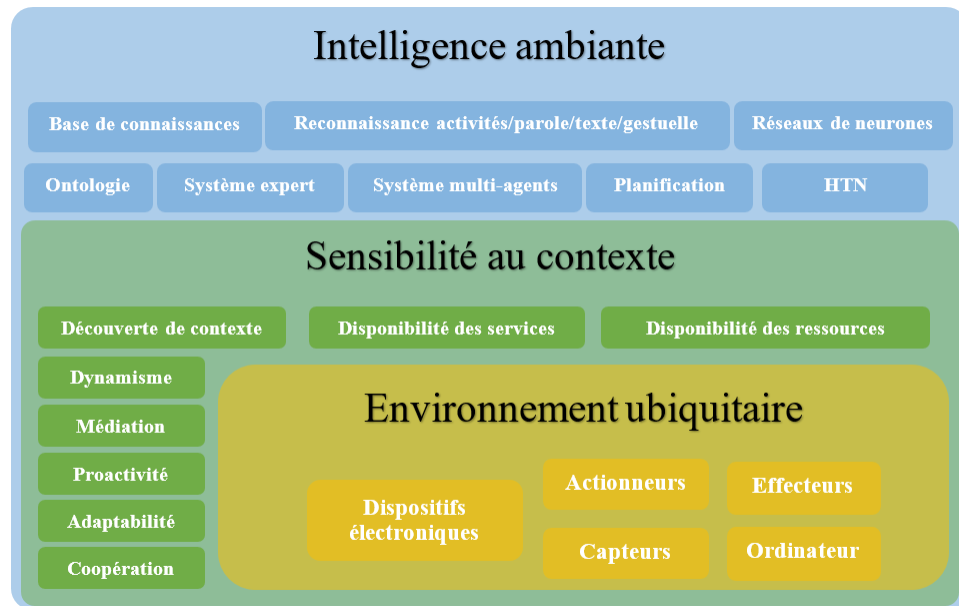


Figure 1.4 – Hiérarchie ubiquitaire - Sensibilité au contexte – Intelligence Ambiante

La Figure 1.4 montre une organisation structurale des domaines mis en place qui forment l'intelligence ambiante. Le bloc environnement ubiquitaire fait référence aux dispositifs électroniques nécessaires à la collecte des informations et à la notification des changements aux utilisateurs. La sensibilité au contexte se sert de ce bloc pour adapter au mieux l'environnement aux besoins de l'utilisateur. La logique et les algorithmes sont mis en place par l'intelligence ambiante.

1.6.3 Les domaines d'applications de l'intelligence ambiante

La démocratisation des composants électroniques permet de faciliter l'épanouissement des conditions de vie des personnes. Plusieurs secteurs sont ainsi touchés par les améliorations proposées. Voici quelques domaines bénéficiant le plus de ce changement de paradigme : les maisons intelligentes, les villes intelligentes, l'assistance médicale et les technologies d'assurances aux personnes ayant des troubles cognitifs, l'hôpital intelligent, les transports intelligents, les services d'urgences, l'éducation, les espaces collaboratifs.

Avant de nous intéresser aux habitants intelligents, nous allons présenter quelques

1.6. INTELLIGENCE AMBIANTE

travaux sur l'AmI disponibles dans la littérature.

1.6.4 L'intelligence ambiante dans la littérature

Les travaux menés dans le domaine de l'intelligence ambiante utilisent différentes approches de l'intelligence artificielle pour prendre en considération les besoins des utilisateurs. Dans cette section, nous nous intéressons à la réalisation de travaux qui ont utilisé une approche sémantique pour la réalisation ou la construction de leur solution d'intelligence ambiante. Chaque solution vise à résoudre un problème spécifique.

L'un des premiers travail dans le domaine est l'ontologie SOUPA [24, 23]. Ce travail modélise le contexte et la représentation des informations de contexte dans une ontologie. Les auteurs mettent en place des mécanismes pour détecter et résoudre des informations de contexte incohérentes et incomplètes dues à une mauvaise collecte. Des mécanismes de partage d'information de contexte, de raisonnement et d'inférence sont utilisés pour l'intégration d'agents intelligents dans la solution proposée. Cependant, trop peu de place est accordée aux personnes.

Pour permettre aux utilisateurs non experts d'accomplir des tâches non triviales et faire face à la complexité croissante des environnements intelligents, les auteurs Golmand et al [45] proposent de déplacer l'attention et les compétences des utilisateurs vers l'essentiel des besoins. En effet, ils considèrent que le système devrait se concentrer en priorité sur l'accomplissement des tâches par l'utilisateur, plutôt que de s'attarder sur les spécifications de comment le système fonctionnera. Pour y parvenir, les auteurs mettent en place un modèle de calcul de tâches offrant aux utilisateurs une vue abstraite des ressources pouvant être utilisées de manière ad hoc. Cette approche centrée utilisateur permet d'implémenter la découverte, la composition et l'exécution des services autour des besoins de l'utilisateur. Pour permettre à l'utilisateur de définir les tâches à un niveau sémantique, une API fait le pont entre la couche d'exécution et la couche sémantique. Cette API est basée sur les services SOAP, WSDL et les technologies UPnP. Les questions de privilèges d'accès et de disponibilité des ressources n'ont pas été traitées dans cette approche. La gestion de contexte n'a pas non plus été abordée.

1.6. INTELLIGENCE AMBIANTE

Pour mettre en place une politique d'accès aux ressources et de déterminer les acteurs, les fonctions et les méthodes, Toninelli et al [110] ont mis en place un modèle de contrôle d'accès. Ce modèle à double enjeu utilise la notion de sensibilité au contexte pour contrôler l'accès à une ressource spécifique. Chaque changement permet d'activer dynamiquement la bonne stratégie d'accès au contexte suivant les changements survenus. Un autre enjeu est celui de la spécification de haut niveau du contrôle d'accès et le raisonnement sur le contexte via des technologies sémantiques. Des ontologies modélisent, infèrent et décrivent le contexte, la sensibilité au contexte, les politiques de contrôle d'accès et leurs adaptations aux changements. Les auteurs ont développé un modèle de politique de contrôle d'accès exploitant la sensibilité au contexte pour la spécification et l'évaluation des politiques définies par les utilisateurs. Le modèle proposé permet la prise en compte du contexte via une combinaison de modélisation ontologique et de programmation logique pour surmonter les limites du raisonnement natif de l'ontologie classique.

Olivares et al [79] illustrent grâce à des personas (personne fictive dotée d'attributs et de caractéristiques sociales et psychologiques et qui représente un groupe cible.) comment leur modèle peut conduire à une assistance préventive efficace pour aider à la préparation de repas. Dans ces travaux, les auteurs souhaitent favoriser l'autonomie des personnes présentant des déficits cognitifs tout en assurant leur sécurité durant le processus de préparation de repas. Un assistant technologique appelé COOK constitué de trois systèmes indépendants est développé. Il permet de guider les utilisateurs durant le processus en lui prodiguant des conseils. Il fournit également à chaque instant des informations et des conseils sur les comportements sécuritaires dans la cuisine. Enfin, il intègre un système de sécurité autonome capable en tout temps d'assurer la sécurité en arrêtant le four en cas de danger ou de prévention contre les dangers.

Pour offrir une représentation formelle des connaissances et partager la logique sous-jacente basée sur le raisonnement de contexte, Wang et al [113] ont développé une ontologie de haut niveau. Cette ontologie, nommée CONON, décrit les concepts sur le contexte et fournit une extension permettant l'ajout d'ontologies de domaines plus spécialisées. Les objectifs de ces travaux étaient de faciliter la représentation de contexte, le partage de contexte, la réutilisation de contexte et d'offrir l'interopéra-

1.6. INTELLIGENCE AMBIANTE

bilité sémantique entre les systèmes hétérogènes. Défini autour de quatorze classes constituant son vocabulaire de base, CONON modélise des personnes, des lieux, des activités, et des entités informatiques. Il faut ajouter à cela la classe de contrainte de qualité utilisée pour représenter différents aspects de la qualité comme la précision et la fraîcheur.

Pour la gestion de la prise en compte du contexte et l'interopérabilité contextuelle, l'ontologie CoOL [107], basée sur le modèle contexte d'échelle d'aspect, implémente la définition d'un vocabulaire plus abstrait et global. L'ontologie agit ainsi comme une couche d'interopérabilité et de comparaison pour permettre des transferts entre des modèles de contextes arbitraires. Semblable au modèle CoOL, [43] propose un méta modèle de contexte qui met en évidence la sémantique des entités, des propriétés et des classes. Ce méta modèle définit des classes de structure de données et des règles de transformations d'une structure de données vers une autre. Chaque modèle de contexte est instancié via OWL-DL pour traiter des prédicats binaires et les informations de contextes sont dérivées via des règles définies dans SWRL. MySAM [19] est un modèle ontologique capable de définir des prédicats de contexte arbitraire. L'élément principal dans ce modèle est la description d'un contexte par des attributs capables d'exprimer des instructions arbitraires d'un domaine contextuel donné. Ce modèle permet de faire la distinction entre une ontologie de contexte et une ontologie de domaine. L'ontologie de domaine est utilisée pour définir les concepts relatifs à un domaine d'application spécifique. L'ontologie du contexte contient tous les attributs de contexte qui s'appliquent aux concepts du domaine.

Dans les travaux [107, 15], les auteurs font un état de l'art de quelques solutions de modélisation de contexte existantes. On y trouve des modèles avec des structures de données simples basées sur des paires clés-valeurs. Principalement, les modèles étudiés par Strang [107] sont organisés en quatre catégories.

1. Les modèles où chaque information de contexte est associée à une valeur, un exemple de tels systèmes est décrit dans [97] ;
2. Des modèles avec des structures de données hiérarchiques utilisant des balises contenant des attributs. Dans ce modèle, la définition des balises est généralement récursive et permet de définir de nouvelles balises [48] ;
3. Des modèles utilisant des représentations graphiques comme le UML sont aussi

1.6. INTELLIGENCE AMBIANTE

étudiés. Certains modèles présentés utilisent les paradigmes de la programmation orientée objet telles l’encapsulation, la réutilisation pour modéliser le contexte.

4. Les modèles logiques et ontologiques sont également étudiés. Ils mettent principalement en avant leur capacité à conclure sur la base de faits et l’inférence depuis un sous ensemble d’informations.

Face aux défis de description du contexte posés par les espaces intelligents dans le développement d’application sensible au contexte et face à la diversité des technologies utilisées, Wang et al, en 2004 [114], ont développé trois axes majeurs pour prendre en considération ces problématiques. Le premier axe est le partage d’une représentation commune et standard du contexte dans un format OWL. Le deuxième axe est celui de requête du contexte. Ainsi, en conservant plusieurs contextes l’accès aux applications est autorisé de manière sélective à un sous-ensemble du contexte. Le dernier axe est celui du raisonnement sur le contexte. Il s’agit ici d’utiliser des informations de haut niveau pour inférer sur la connaissance de l’état de l’utilisateur et de son entourage. La structure hiérarchique des ontologies proposées permet aux développeurs de réutiliser les ontologies de domaine (par exemple, des personnes, des périphériques et des activités) pour décrire des contextes et créer un modèle de contexte pratique sans recommencer à zéro chaque fois.

Pour travailler avec les perspectives cognitives et comportementales, Santofimia et al [96] se proposent de mettre en place un modèle sémantique d’intelligence ambiante qui décrit les actions, les événements et leur interconnexions. Le modèle proposé met en place un moteur de raisonnement et une modélisation contextuelle des événements pour générer des réponses comportementales adéquates. Les éléments abordés dans cette étude sont l’autogestion, la proactivité, le dynamisme et le comportement axé sur les objectifs.

Pour guider le processus de conception et intervenir au plus tôt, Diamantini et al [35] ont construit un système d’assistance domotique en présentant une méthodologie axée sur les ontologies et destinée à guider le processus de développement. Le but est d’analyser et de représenter les exigences des systèmes d’assistance dans une ontologie de haut niveau. Chacune des exigences est décrite sous forme de sous-objectifs et de tâches reliées à des mesures. La démarche multidisciplinaire tient compte de

1.6. INTELLIGENCE AMBIANTE

l'analyse des exigences et des traces de relations entre chaque exigence pour proposer des solutions adaptées aux objectifs. Une représentation formelle des objectifs, la décomposition en tâches à accomplir et la référence à un langage logique pour la représentation des connaissances au moment de la conception constituent la solution proposée.

Toujours pour répondre aux défis de contexte imparfait et à la nature dynamique et hétérogène des environnements ambiants, Bikakis et al en 2008 [15] proposent une ontologie pour la gestion des connaissances en intelligence ambiante. Cette ontologie permet de raisonner avec des données de contexte dynamiques et ambiguës, de gérer en temps réel les données contextuelles potentiellement volumineuses, compte tenu des capacités de calcul restreintes de certains appareils mobiles, et de créer de l'intelligence collective, en soutenant le partage d'information et le raisonnement distribué entre les entités de l'environnement ambiant.

Pour prendre en considération les problèmes d'hétérogénéité, de représentation des données et du fait que les nombreux modèles existants ne sont pas capables de prendre en compte différents aspects d'expressivité des énoncés (arbitrage des prédicats, prédicats bien structurés), Sorici et al [102] propose CONSERT. Cette approche de méta modélisation de contexte offre un moyen cohérent et uniforme de travailler avec, d'une part, les experts du domaine et, d'autre part, avec les contraintes et méta propriétés liées à celle-ci. Ils proposent une formalisation du modèle et une implémentation qui utilisent les ontologies et SPARQL (SPARQL Protocol and RDF Query Language) .

Basé sur le paradigme de multi contexte, Bikakis et al, en 2010 [16], utilisent un raisonnement logique avec des caractéristiques logiques non monotones couplées à un mécanisme d'ordonnancement des préférences pour gérer les informations de contextes incertaines et conflictuelles. La non-monotonie permet de définir des règles locales facilitant la sensibilité au contexte. Des approches dirigées par des inférences proposent l'utilisation d'ontologie pour modéliser des activités complexes en utilisant les requêtes SPARQL. Le raisonnement porte sur la composition d'activités du simple au plus complexe [75].

Pour apporter une solution aux limites des approches fondées sur les règles et surtout sur l'utilisation des théories d'actions (étude de la nature de l'action humaine),

1.7. SYNTHÈSE DE L'ÉTAT DE L'ART

Patkos et al [82] conçoivent un Framework de raisonnement. Le but est d'appliquer à une infrastructure d'intelligence ambiante opérationnelle une ontologie pour la représentation des connaissances contextuelles. Le raisonnement effectué dans ce cas est fondé à la fois sur des règles, sur des actions et sur leur causalité. Ce raisonnement permet de prendre en charge une multitude de tâches afin de combler l'écart des lacunes de certaines actions.

1.7 Synthèse de l'état de l'art

La revue de la littérature a permis de présenter les concepts et les définitions nécessaires à la compréhension du domaine de l'intelligence ambiante. Les solutions proposées s'appuient sur des techniques et des outils offerts par l'intelligence artificielle. Ces techniques utilisent le contexte de l'utilisateur pour assurer une meilleure utilisation des ressources ainsi qu'une meilleure utilisation des informations ambiantes.

L'intelligence ambiante désigne la capacité pour un environnement donné, comprenant des personnes et des dispositifs électroniques embarqués, d'être capable de détecter leur état, d'anticiper et de s'adapter éventuellement aux besoins de ces personnes. Il en ressort que la mise en place d'un tel système nécessite d'avoir une compréhension précise sur l'environnement, ses spécificités, ses contraintes et comment il interagit avec les autres composants. Un autre niveau de compréhension sur la structuration des informations de la personne est nécessaire, c'est-à-dire comment elle fait pour réaliser un besoin, comment se déclinent les étapes, comment pourrait-on l'aider, quel mécanisme mettre en œuvre pour structurer son profil, comment s'organiser pour valider la cohérence, la conformité et la précision des besoins. Il convient d'ajouter à cela qu'une autre compréhension sur l'utilisation judicieuse des dispositifs électroniques pour s'assurer d'offrir une meilleure assistance à la réalisation des besoins de la personne est requise.

Fondamentalement, il s'agit de comprendre, de structurer, de formaliser et de mettre en évidence les relations existantes entre quatre niveaux d'informations : l'environnement (habitat), les dispositifs technologiques (capteurs, effecteurs...), les personnes et l'assistance à offrir. Bâtir ces connaissances nécessite l'utilisation d'outils capables d'offrir un haut niveau d'expression et de représentation de la connaissance,

1.8. CONCLUSION

d'où l'utilisation des ontologies. La revue de la littérature présente des solutions se basant sur les ontologies pour offrir de l'assistance ambiante. Celles-ci ont pour la plupart été construites dans une logique de représentation d'un ou d'un sous-ensemble de niveau d'information. Aucune des solutions proposées ne permet d'encapsuler la totalité des quatre niveaux d'informations que nous avons identifiés. Ce manquement ne permet pas de structurer la totalité de la connaissance sémantiques existantes entre différents concepts de l'intelligence ambiante telle que nous la concevons.

Pendant le processus d'analyse et de structuration de la connaissance autour de la réalisation des besoins utilisateurs, deux préoccupations ont miné notre raisonnement. Qu'est-ce qu'un scénario et comment valide-t-on sa cohérence ? L'ontologie permet de le spécifier, mais il intervient trop loin dans le processus de conception et sa validation ne se fait qu'au niveau de l'implémentation. Nous voulons une démarche qui permet d'agir au plus tôt dans le processus d'analyse et de conception afin de permettre à l'équipe multidisciplinaire de rester concentrée sur les objectifs et l'analyse et non sur les détails d'implémentation. Par raffinements successifs, nous vérifions de manière globale que les spécifications fonctionnelles d'un scénario donné restent cohérentes et ne créent pas d'ambiguïté.

1.8 Conclusion

L'utilisation des ontologies comme outil de modélisation de contexte dans la littérature offre une richesse de structuration, de description et de raisonnement sur les informations du contexte permettant d'améliorer l'AmI. La revue de littérature montre que très peu de solutions mettent en symbiose la totalité de la connaissance ambiante dans laquelle sont constamment plongés les utilisateurs. De plus, seules quelques études modélisent le contexte dans un esprit DIY. Le choix des ontologies comme outils de modélisation sémantique met en place un cadre visant à rendre le contenu des ressources accessible et utilisable par les programmes et agents logiciels, grâce à un système de métadonnées formelles. L'ontologie contrairement à UML/EA fonctionne dans une hypothèse de monde ouvert c'est-à-dire que les données manquantes sont considérées comme inconnues.

Chapitre 2

Les Habitats Intelligents

L'intelligence ambiante offre aux habitats intelligents l'analyse du contexte. Ceux-ci sont ainsi conscients des besoins de ses habitants, des exigences de la personnalisation et des comportements ou de tout autre contexte pouvant survenir. Dans cette section, nous allons présenter les solutions de maisons intelligentes existantes. Nous parlons aussi des technologies et des mécanismes à mettre en place pour collecter les informations dans un habitat afin de le rendre intelligent.

2.1 Des habitats automatiques aux habitats intelligents

Dans la littérature, le terme maison automatisée (*Home automation*) [92] désigne un ensemble de solutions utilisées pour contrôler, surveiller, gérer et surtout automatiser certaines fonctions dans la maison. Les dispositifs qui sont uniquement contrôlés par des interrupteurs, des moteurs, ou des retardateurs horaires ne font pas partie de cette définition. La démocratisation des objets connectés et l'utilisation des nouveaux protocoles de communications sans fil IEEE 802.15 ont permis une évolution du terme maison automatisée vers habitat intelligent ou maison intelligente, appelé en anglais « *Smart Home* » [120, 50, 12, 120, 50, 92, 36, 112, 99, 6, 26].

Une maison intelligente est une résidence équipée de dispositifs technologiques qui observent et collectent les informations sur le résidant afin de lui offrir de manière

2.1. DES HABITATS AUTOMATIQUES AUX HABITATS INTELLIGENTS

proactive, des services adaptés à ses besoins. Dans la suite de ce document nous utilisons indépendamment le terme habitat intelligent ou smart home pour faire référence à la même entité.

Les habitats intelligents sont donc l'application de l'intelligence ambiante à la gestion des AVQ et au bien-être des personnes. Plus généralement, ils sont considérés comme des domaines d'application de l'intelligence ambiante. Dans sa thèse, Gallisot [44] rappelle les quatre concepts permettant de discriminer maison non intelligente et maison intelligente. Une maison intelligente se caractérise donc par le fait que :

1. L'information ne peut pas circuler librement de l'intérieur de la maison vers le monde extérieur, et vice-versa ;
2. La maison fonctionne avec des ordinateurs intégrés qui ne peuvent pas se parler entre eux ;
3. L'automatisation de la maison consiste en un « patchwork » de gadgets ;
4. La maison est équipée avec des fonctions sophistiquées difficiles à utiliser.

De ce constat, et selon la définition précédemment donnée, six types mettent en exergue les fonctionnalités des habitats intelligents.

1. Gestion de l'énergie et contrôle du climat ambiant de l'habitat. Il s'agit de définir des politiques d'optimisation de la gestion énergétique. Prévoyance, adaptation, optimisation sont les éléments clés qui entrent dans ce processus.
2. Système de contrôle de l'éclairage et gestion de la mise en fonction des ouvertures. L'idée est d'offrir un certain niveau de confort pour faciliter la qualité de vie. La maison intelligente s'adapte constamment à son environnement pour automatiser des tâches de maintenance, d'entretien, d'allumage électrique de façon transparente.
3. Gestion de la sécurité et du contrôle des accès. Pour prévenir les accidents, les chutes, et les vols, les habitats intelligents émettent des notifications sur les consignes de sécurité et des alertes d'intrusion vers les personnes ou les proches en cas de risques. Certains systèmes offrent des supports pour la compréhension de la réaction du système face au problème survenu pour un meilleur suivi.
4. Gestion des appareils ménagers. Généralement mis en place pour faciliter le partage des ressources entre les habitants d'une propriété. L'habitat intelligent

2.2. LES TECHNOLOGIES POUR LES HABITATS INTELLIGENTS

émet des notifications lorsqu'une ressource (ex. machine à laver) est en cours d'utilisation. Le scénario qui revient le plus fréquemment est celui d'une personne âgée qui parcourait quatre ou cinq étages pour laver son linge au sous-sol alors que la machine à laver est encore en utilisation.

5. Système audiovisuel et de gestion des divertissements. Dans le but d'améliorer le confort des personnes, les habitats intelligents reconnaissent la personne et selon ses goûts, diffusent une ambiance personnalisée pendant sa présence.
6. Système d'assistance aux activités de la vie quotidienne, maintien de personnes âgées à domicile et support et assistance aux personnes ayant des déficits cognitifs. Il s'agit de mettre en place une assistance permanente pour aider les personnes à réaliser certains AVQ.

Dans la suite de notre document, nous nous intéressons uniquement à la sixième catégorie.

2.2 Les technologies pour les habitats intelligents

Les solutions basées sur la transmission sans fil, l'infrarouge et les courants porteurs sont principalement utilisées pour offrir des services de communications de courte distance au sein des habitats intelligents. Ils ont facilité l'acceptation de la domotique dans les habitats intelligents. La démocratisation des protocoles de communication 802.15.x a facilité la communication et le partage de données entre diverses entités. Dans cette section, nous allons énumérer les technologies les plus marquantes conformément à leur standard ouvert et les facilités offertes en intégration avec d'autres logicielles : X10, Insteon, Z-Wave, ZigBee, Chacon et enOcean. Nous parlerons des deux technologies les plus répandues.

Z-Wave [9], développé par Zensys en 2004, utilise un protocole de communication sans fil pour envoyer des commandes à des dispositifs électroniques. Théoriquement, le protocole supporte plus de 232 nœuds. Les dispositifs sont bidirectionnels et communiquent entre eux pour couvrir une plus grande surface. La compatibilité est assurée entre les différents constructeurs. Pour fonctionner, le protocole a besoin d'un contrôleur principal dont le rôle est de créer le réseau Z-Wave par ajout consécutif

2.3. LES SOLUTIONS D'HABITATS INTELLIGENTS EXISTANTES

de dispositifs. Cette construction itérative permet de désigner certains nœuds comme répéteur pour faciliter la propagation des signaux et la réduction des interférences. De nos jours, elle se décline dans sa version Z Wave+.

La technologie **ZigBee** [66], ou IEEE 802.15.4, permet de définir un réseau sans fil à très bas prix, de faible portée et avec une faible consommation d'énergie. Cette technologie permet d'obtenir des débits pouvant atteindre 250 kbps avec une portée maximale d'environ 100 mètres. Ce protocole permet la transmission de données sans fil, machine à machine. Cette technologie est le prolongement de la norme HomeRF (Home Radio Frequency). Les nœuds sont organisés en graphe avec trois rôles :

1. Un contrôleur, appelé *ZigBee Controller*, responsable du réseau, du routage et des opérations de gestion et du maintien de groupe.
2. Des nœuds routeurs, appelés *ZigBee Router*, qui se comportent comme des relais de transmissions d'informations.
3. Enfin des nœuds terminaux, appelés *ZigBee End Device*, qui ne font pas de routage, mais envoient au contrôleur ou reçoivent du contrôleur uniquement des informations.

2.3 Les solutions d'habitats intelligents existantes

Plusieurs projets de recherche se fondent sur les laboratoires vivants pour concevoir et offrir une vaste gamme de solutions de maisons intelligentes. Un laboratoire vivant est un laboratoire qui regroupe dans une optique de multidisciplinarité des représentants de l'industrie, des chercheurs et des universitaires qui travaillent ensemble pour offrir plus de recherche innovante. Dans l'industrie, la tendance est plutôt à la conception d'outils permettant de transformer une maison quelconque en maison intelligente. La suite de cette section offre un tour rapide de quelques solutions existantes de maisons intelligentes.

2.3.1 Les solutions de laboratoires de recherche

Nous présenterons dans cette section six laboratoires, qui utilisent des approches différentes pour offrir des services d'intelligence ambiante.

2.3. LES SOLUTIONS D'HABITATS INTELLIGENTS EXISTANTES

Le DOMUS

Comment offrir une reprise d'autonomie aux personnes âgées ou personnes avec traumatisme crânien quand elles présentent des troubles cognitifs ? Telle est la question à laquelle le laboratoire DOMUS de l'Université de Sherbrooke se consacre depuis 2002. Ce laboratoire utilise une approche multidisciplinaire pour offrir, dans un habitat intelligent installé au sein du campus de l'Université de Sherbrooke, des services adaptés permettant d'accompagner des personnes dans la réalisation des AVQ (ménage, hygiène, prise de médicaments, préparation de repas. . .) [54, 53, 51, 99].

Smart Rooms

Mis en place dans les années 1996 par Pentland du MIT (Massachusetts Institute of Technology) Media Lab, «Smart Rooms» [83] est considéré comme étant parmi les premiers laboratoires d'habitat intelligent à offrir des services d'assistance. Les auteurs ont principalement développé dans ce laboratoire un ensemble de programmes informatiques nécessaires à la reconnaissance des expressions faciales et des gestes. Leur approche utilise des caméras et des micros pour reconnaître la personne, ses activités, sa localisation, sa voix et ses activités de conduite auto.

Place Lab

«Place Lab» [60] est l'habitat intelligent du projet «House_n» [59] du MIT, dont l'objectif est de bâtir la maison du futur. Ce projet sert de validation et de preuves de concepts pour de nombreux travaux liés à l'intelligence ambiante dans le laboratoire. La complexité des services d'assistance offerte aux AVQ a transformé le laboratoire de maison intelligente en laboratoire pluridisciplinaire. Le projet House_n vise à encourager les comportements sains par l'utilisation des techniques d'interaction personne-machine, à offrir une assistance aux personnes âgées via la reconnaissance d'activités et à finalement intégrer dans les activités courantes des mécanismes de surveillance et d'assistance médicale via la surveillance biométrique du rythme cardiaque, de la tension artérielle et de certains paramètres physiologiques.

2.3. LES SOLUTIONS D'HABITATS INTELLIGENTS EXISTANTES

CASAS

Pour piloter son environnement à l'aide de capteurs et effecteurs les chercheurs du projet CASAS [90] ont modélisé l'habitat comme un agent intelligent qui a des buts, qui perçoit son environnement et le pilote. L'objectif étant l'amélioration du confort des personnes et la diminution des coûts de fonctionnement énergétique. La maison intelligente du projet construit au sein du campus de Washington State University fonctionne sur un principe d'apprentissage des habitudes de l'habitant et non sur la reconnaissance d'activités [62, 8, 22].

The intelligent Dormitory (iSpace)

Situé sur le campus de l'Université d'Essex, iSpace [56] est un appartement intelligent capable d'enregistrer des informations relatives à l'habitant pour des besoins d'apprentissage. En utilisant les techniques de logique floue [37], les auteurs proposent une solution basée sur les agents pour adapter l'assistance au comportement appris. Ainsi par minimisation de la bande passante, par optimisation du confort et par minimisation de la consommation d'énergie, ils proposent une optimisation multi objectif pour la coordination des dispositifs hétérogènes utilisés dans l'appartement intelligent [37, 109].

HomeCare

Face aux problèmes de perte d'autonomie des personnes âgées souffrant de la maladie d'Alzheimer, les responsables du laboratoire LAAS de Toulouse ont mis en place un système de surveillance basé sur des caméras appelé «HomeCare» [21]. Grâce aux données collectées et aux analyses effectuées sur l'état de santé et la façon dont se déplace la personne, ils sont capables de détecter des risques de chutes. Les approches d'apprentissage automatique basées sur les réseaux de neurones sont utilisées pour générer des notifications lorsqu'il y a des agitations nocturnes ou autres situations ne respectant pas les conditions initiales.

2.3. LES SOLUTIONS D'HABITATS INTELLIGENTS EXISTANTES

2.3.2 Les solutions industrielles

Plusieurs offres industrielles sont disponibles. Deux principales solutions se dégagent : les solutions clés en main et les solutions sous forme de boîtes domotiques à installer soi-même.

Les solutions clés en main

Généralement, il s'agit de laboratoire industriel utilisé pour tester les solutions à proposer aux personnes.

«The TRON House» à Tokyo au Japon : Considéré comme l'une des premières maisons intelligentes, le TRON House (The Realtime Operating system Nucleus) [10] faisait tourner un très grand nombre d'applications sur des centaines de processeurs intégrés dans des objets à l'intérieur d'un habitat afin d'offrir plus de confort. Cette solution fut particulièrement intrusive (installation massive d'équipements) et aucune solution de mise à l'échelle n'ayant été pensée, elle fut abandonnée par la suite.

La «Toyota Dream House» : La «Toyota Dream House» [11], succède à la première version de the «TRONE» et s'appuie sur les concepts d'intelligence ambiante. La nouvelle maison offre plus de confort et de facilité pour la gestion de son habitat et du garage automobile. En effet, la maison collecte de l'énergie pour alimenter une voiture hybride qui à son tour alimente l'ensemble du système en cas de coupure d'énergie. La plupart des objets sont équipés par la technologie RFID.

Les boîtes domotiques

Plusieurs solutions permettant de contrôler le chauffage, la lumière, les volets et certains services à la maison existent. Le marché de la domotique s'ouvre donc ainsi aux particuliers en limitant les interactions avec les experts (Techniciens du domaine). En effet, les solutions proposées sont fournies avec des manuels d'installation et plusieurs vidéos permettent de faciliter leur compréhension. Il existe une large gamme de choix domotiques. L'utilité de la boîte est de permettre à chacun de transformer sa

2.4. CONCLUSION DU CHAPITRE

maison classique en maison intelligente. Lesquelles que solutions de boîte domotiques existantes sont :

- «Fibaro Home Center» est une boîte supportant le protocole Z-Wave avec une compatibilité à plusieurs dispositifs électroniques.
- «Vera Lite» est une boîte qui offre un contrôle et une compatibilité avec environ 750 dispositifs de plusieurs types.
- «Zipabox» boîte qui offre une compatibilité avec plusieurs protocoles 802.15.4.
- «Samsung SmartThings» est une boîte qui permet de connecter environ 200 appareils.
- «Amazon Echo» ou encore «Alexa» est une boîte capable de piloter une maison ou un ensemble de dispositifs grâce à la voix.
- «HomeKit» est une boîte domotique proposée par Apple pour piloter une maison.
- «Google Home» est un haut-parleur qui permet de commander par l’assistant vocal de Google des objets intelligents ou une maison intelligente.

2.4 Conclusion du chapitre

Du fait de la diversité des outils et des solutions techniques proposées, les habitats intelligents permettent de prendre en considération les besoins précis et les personnalisés des utilisateurs. Des tâches telles que de la prise de médicaments à la préparation des repas en passant par la gestion automatique de l’éclairage, de nombreuses techniques ont été développées pour simplifier la réalisation par les utilisateurs de leurs besoins. Le développement de l’intelligence ambiante a permis la création d’un nouveau domaine visant à favoriser le maintien à domicile des personnes âgées. L’assistance ambiante au maintien à domicile des personnes âgées (*Ambient Assisted Living*) permet aujourd’hui de créer des conditions favorables à l’émergence et à la compréhension des facteurs nécessaires à la création d’un bien-être chez les personnes âgées.

Deuxième partie

Contributions

Chapitre 3

Spécification Formelle de Scénarios

Dans ce chapitre, nous modélisons et spécifions le comportement des AVQ via des scénarios. L’objectif est de mettre en place une démarche formelle pour enlever toutes ambiguïtés afin de spécifier de manière logique les contraintes issues de la réalisation des AVQ. La modélisation proposée permet d’assurer durant tout le processus de co-conception la conformité et la validité du scénario.

Le langage ALLOY [61] est utilisé pour spécifier et prouver l’exactitude et la conformité des scénarios. La conception dirigée par les scénarios permet à l’utilisateur final de s’assurer que l’exécution ou la réalisation de ses besoins respectent ses propres contraintes. Les spécifications formelles sont utilisées pour prouver l’exactitude des systèmes critiques. De notre point de vue, il nous semble que l’assistance apportée via l’intelligence ambiante aux personnes âgées est assez critique. Un article scientifique a été publié pour décrire nos travaux [65].

La suite de ce chapitre propose une approche pour la création, la spécification, la modélisation et la validation automatique des scénarios durant le processus de conception. Après avoir défini ce qu’est un scénario, nous donnons un exemple de scénarios. Le paragraphe suivant part de cet exemple pour bâtir un ensemble de spécifications formelles permettant de valider l’écriture et la conception des scénarios. La partie sur le formalisme décrit la spécification produite pour décrire les scénarios et l’utilisation de l’analyseur ALLOY. Cet analyseur détermine de manière incrémentale avec des simulations si les spécifications restent conformes et valides au modèle lorsque la taille augmente de manière considérable. La modélisation proposée s’inspire du

3.1. DÉFINITION DES CONCEPTS CLÉS

modèle de représentation des tâches de haut niveau : le modèle CTT (Concur Task Trees) [77] et du modèle hiérarchique d'analyse des tâches HTA (Hierarchical Task Analysis) [11].

3.1 Définition des concepts clés

Pour mettre en place un système d'assistance aux AVQ, les différents experts du domaine et toutes les parties prenantes intervenant dans le processus ont besoin de procédés itératifs, incrémentaux et inclusifs. Pour que le système soit sensible au contexte l'utilisateur doit être au cœur du processus et toute la démarche doit être centrée utilisateur. Pour éviter toutes ambiguïtés, nous avons pris pour résolution d'adopter les définitions ci-dessous. Elles sont celles qui correspondent le mieux de nos exigences.

3.1.1 Définition : Scénario

Un scénario décrit l'agencement des activités à réaliser pour atteindre un but. L'approche de scénario constitue une forme de réflexion sur la conception participative et multidisciplinaire. L'utilisation des scénarios implique de se poser plusieurs questions entre autres : Comment va-t-il se présenter ? Sous quelle forme ? Avec quels instruments ? Avec quel degré de fiabilité ? Des exemples de scénarios sont : errance nocturne ; Boire de l'eau ; médication ; préparation d'un repas.

3.1.2 Définition : Scénario d'assistance

Un scénario d'assistance est un scénario qui rajoute à la définition du scénario la description du comportement du système, le comportement et le profil de l'environnement physique (habitat) et les interactions entre toutes les entités impliquées. C'est donc une courte histoire sur les productions à réaliser ainsi que les règles et les consignes d'intervention auprès de toutes les parties prenantes.

3.1. DÉFINITION DES CONCEPTS CLÉS

3.1.3 Définition : Activité

Une activité est un ensemble ordonné ou non de comportements ou de tâches. On distingue deux types d'activités.

1. Les activités attendues décrivent un ensemble de comportements pouvant être réalisés pour atteindre un but.
2. Les activités réalisées sont un ensemble de comportements observés conduisant à la réalisation d'un but.

Quelques exemples sont : trouver un verre pour boire de l'eau ; sortir de son lit pour se rendre aux toilettes...

3.1.4 Définition de tâches

Une tâche est une procédure établie permettant la réalisation d'une activité. Elle peut se décomposer en sous-tâches. Des exemples de tâches sont : trouver un verre pour boire de l'eau ; ouvrir l'armoire qui contient les verres ; se laver les mains...

3.1.5 Définition d'actions

Une action est une tâche terminale c'est-à-dire une tâche ne pouvant pas se décomposer en sous-tâches. Elle est issue d'un changement d'état ou de la réponse à un changement d'état. Plus généralement, c'est un état pouvant être compris et détecté par un capteur. Des exemples de tâches sont : prendre le verre, ouvrir le robinet, se déplacer, boire...

3.1.6 Définition de conception dirigée par les scénarios

La conception dirigée par les scénarios est un ensemble de méthodes relativement légères qui mettent l'accent sur la définition concrète des activités à réaliser au début du processus de développement. Son avantage par rapport aux autres approches centrées utilisateurs est qu'il se concentre directement sur la manière dont les utilisateurs utiliseront le système plutôt que de se concentrer sur les fonctionnalités qui définissent le système.

3.2. DES EXIGENCES MULTIDISCIPLINAIRES AUX SCÉNARIOS

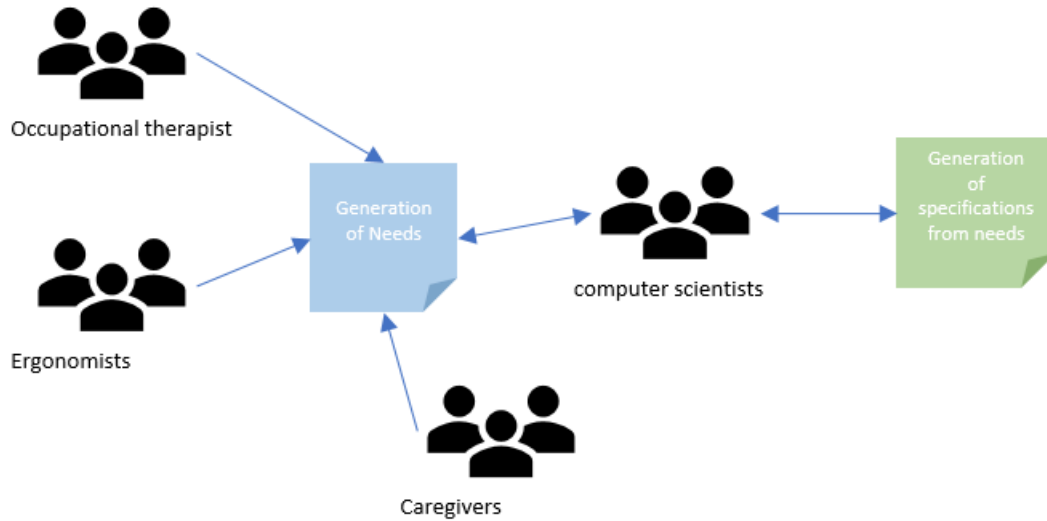


Figure 3.1 – Processus standard pour la production des spécifications utilisateurs [65]

3.2 Des exigences multidisciplinaires aux scénarios

Plusieurs parties prenantes sont impliquées dans le processus de mise en place d’une solution d’assistance ambiante. On y retrouve entre autres les informaticiens, les ingénieurs, les ergothérapeutes, les ergonomes, les gérontologues, les aidants, les aidés etc. Traditionnellement, les exigences sont envoyées sous forme de cahier des charges aux techniciens pour traduction en spécifications et en besoins fonctionnels comme le montre la Figure 3.1. Cette approche ne favorise pas la collaboration multidimensionnelle, en ce sens que seuls les techniciens sont capables de lire les spécifications produites. Ce procédé ne permet pas aux non-experts de comprendre les nouvelles exigences issues des contraintes systémiques. De plus, il ralentit les traitements en augmentant le nombre d’allers et de retours tout en favorisant l’introduction d’erreurs. Les parties prenantes n’ont que des vues partielles du processus à différentes étapes du procédé.

On voudrait offrir la possibilité aux utilisateurs et aux concepteurs, qui veulent progresser rapidement et obtenir des rétroactions sur la conception, de le faire tout en continuant de raffiner leurs idées. Nous proposons d’extraire le contexte et de structurer les informations pertinentes des besoins, afin de bâtir une architecture dans laquelle toutes les parties prenantes ont en tout temps la même vue sur toutes

3.2. DES EXIGENCES MULTIDISCIPLINAIRES AUX SCÉNARIOS

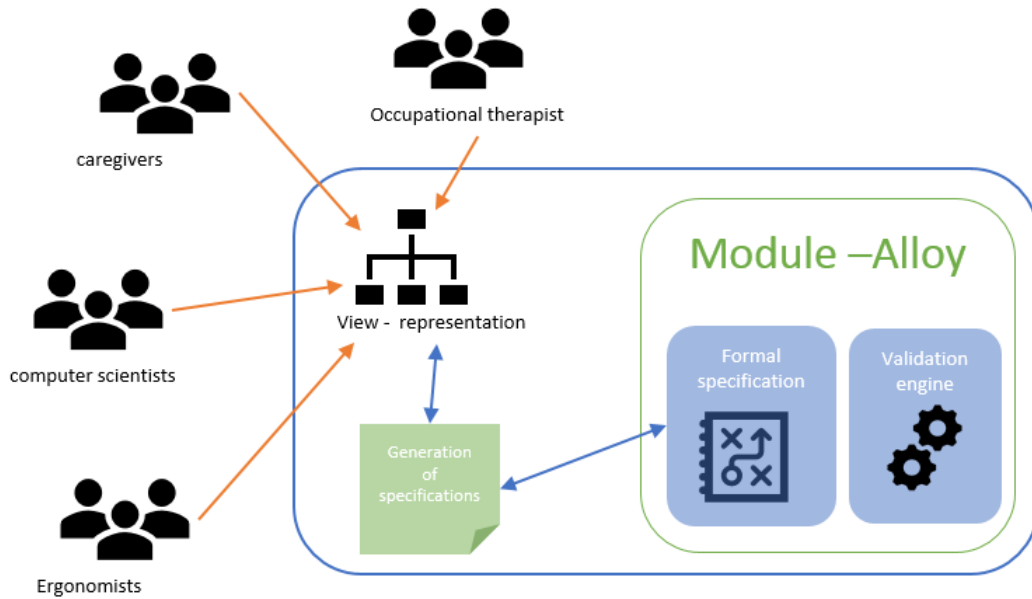


Figure 3.2 – Processus multidisciplinaires de co-conception hiérarchique de scénarios [65]

les étapes du processus. Cette architecture s’organise autour d’un modèle hiérarchique de construction de scénarios. Une validation automatique est faite chaque fois que de nouveaux besoins sont ajoutés.

La validation est un processus utilisé pour vérifier de manière globale que les spécifications fonctionnelles d’un système donné restent cohérentes et ne créent pas d’ambiguïté. L’exactitude est définie ici comme une conformité aux exigences des technologies pour l’assistance. La Figure 3.2 montre notre processus organisationnel. Dans cette figure, la représentation hiérarchique de l’organisation des scénarios permet une lecture facile pour toutes les parties prenantes. Ils partagent tous la même vue à tous les niveaux du processus même quand une nouvelle itération est ajoutée au scénario. Par la suite, des spécifications conformes aux scénarios sont générées automatiquement par le module ALLOY. L’analyseur interne se charge de vérifier que les nouvelles spécifications sont conformes au modèle.

3.3 Caractéristiques du modèle hiérarchique

En utilisant les connaissances et les solutions proposées dans la littérature concernant les HTA [11] nous proposons un modèle hiérarchique d'analyse de tâches. Le choix de ce modèle hiérarchique répond à cinq critères :

1. **Une représentation simple de la connaissance** : ce critère permet de partager plusieurs expertises induites par le caractère multidisciplinaire des membres de l'équipe. Les connaissances non techniques sont représentées sous forme d'arbre avec comme nœuds des activités ou des actions à effectuer. L'attention de toute l'équipe est mise sur les actions de l'utilisateur sans s'inquiéter sur les fonctionnalités du système. Le scénario est construit itérativement durant la période de remue-méninges.
2. **Consensus entre les parties prenantes** : dans ce critère, chaque partie prenante spécifie clairement et partage les contraintes de son domaine. La création de plusieurs niveaux hiérarchiques pour la représentation des tâches permet de créer des catégories consensuelles. Si un nouveau besoin survient, il est facile de créer une indirection qui prend en compte ses particularités.
3. **Recentrer l'attention** : la représentation hiérarchique des scénarios permet de se concentrer sur un objectif précis, sur une tâche précise plutôt que sur une vision globale de la solution.
4. **Spécifications plus concrètes** : une représentation de modèle hiérarchique dans une organisation à plat est plus simple à lire qu'une représentation à plat dans la même organisation. Cette représentation permet de passer d'un modèle concret à un modèle abstrait et vice-versa.
5. **Très faible complexité** : pour atteindre un objectif complexe, on peut le décomposer en sous-objectifs de complexité moindre. Il s'agit du passage de tâche à sous-tâches. Chaque tâche peut être analysée indépendamment et individuellement durant le processus. Plusieurs sous-scénarios peuvent être combinés entre eux pour avoir un scénario plus grand et plus complexe.

3.4. EXEMPLES DE SCÉNARIOS

3.4 Exemples de scénarios

Pour faciliter la compréhension et le développement des spécifications formelles produites, nous avons défini quatre scénarios. Ils ont été utilisés pour le développement de notre modèle.

3.4.1 Scénario 1 : Aller aux toilettes durant la nuit depuis sa chambre à coucher

Il est plus de 23 heures et une envie pressante d'aller aux toilettes se fait ressentir. La personne se réveille pendant la nuit. Quand elle sort de son lit, éclairer la chambre à 10%. Suivant certaines contraintes, on utilisera soit le plafonnier soit une lampe de chevet. Une fois que la personne commence à se déplacer dans la chambre, le chemin lumineux menant vers la salle de bain se trouvant entre le corridor et la chambre s'allume et le plafonnier de la salle de bain s'illumine à 10% une fois que la personne traverse la porte de la chambre, les lumières de celle-ci s'éteignent. À la traversée de la porte de la salle de bain, son plafonnier se met à 25% d'intensité et la toilette s'illumine. Pendant ce temps, le chemin lumineux entre la chambre et la salle de bain s'éteint. Si la chasse d'eau est tirée, le lavabo s'éclaire. Si la personne utilise le lavabo pour se laver les mains, l'intensité de la lumière de la salle de bain est descendue à 10% et le chemin lumineux des toilettes vers la chambre s'allume. Si elle ne se lave pas les mains, et est détectée hors de la salle de bain, la lumière de la salle de bain est descendue à 10% et le chemin lumineux des toilettes vers la chambre s'allume. Une fois la personne détectée dans le corridor, la lumière de la chambre s'allume à 25% et le plafonnier de la salle de bain est éteint. De retour dans la chambre, le chemin lumineux est éteint et la lumière passe à 10%. Lorsque la personne retourne dans son lit, la lumière de la chambre s'éteint.

3.4.2 Scénario 2 : Boire un verre d'eau au robinet dans la cuisine

À une heure très avancée de la nuit, la personne se réveille et a soif. Dès qu'elle sort de son lit, éclairer la chambre à 10%. Suivant certaines contraintes, on utilisera soit

3.4. EXEMPLES DE SCÉNARIOS

le plafonnier soit une lampe de chevet. Lorsque la personne commence à se déplacer dans la chambre, le chemin lumineux menant vers la cuisine se trouvant entre le corridor et la chambre s'allume et le plafonnier de la cuisine s'allume à 10%. Dès que la personne a traversé la porte de la chambre, les lumières de celles-ci s'éteignent. À l'entrée dans la cuisine, son plafonnier passe à 25% d'intensité, le chemin lumineux s'éteint et l'armoire contenant les verres s'illumine. Le filtre varioglass de la partie de l'armoire pour verre à eau devient transparent pour laisser entrevoir son contenu (voir Figure 3.3). À l'ouverture de l'armoire, le robinet de la cuisine s'illumine et les lumières de l'armoire sont éteintes. Le varioglass devient opaque. À la fin de l'utilisation du robinet, attendre environ 3 minutes que la personne ait fini de boire. Dès que la luminosité de la cuisine est réduite à 10%, allumer le chemin lumineux vers la chambre, éteindre le lavabo. Faire la même chose si la personne commence à quitter la pièce. Dès que la personne est détectée, dans le corridor la lumière de la chambre s'allume à 25% et la lumière de la cuisine est éteinte. De retour dans la chambre, le chemin lumineux est éteint et la lumière passe à 10%. Lorsqu'elle retourne dans son lit, la lumière de la chambre s'éteint.

3.4.3 Scénario 3 : Se relaxer au salon

Agitée, très agitée, la personne n'arrive plus à retrouver le sommeil et ne veut plus rester au lit. Quand elle sort de son lit, éclairer la chambre à 10%. Suivant certaines contraintes, on utilisera soit le plafonnier soit une lampe de chevet. Lorsque la personne commence à se déplacer dans la chambre, le chemin lumineux menant vers le salon se trouvant entre le corridor et la chambre s'allume et le plafonnier du salon s'allume à 10%. Lorsque la personne a traversé la porte de la chambre, les lumières de celles-ci s'éteignent. Lors de son entrée dans le salon, le plafonnier passe à 25% d'intensité, le chemin lumineux s'éteint et la musique de l'utilisateur s'affiche sur la télévision du salon. En même temps, une image relaxante est affichée en fond d'écran. La chaise berçante est éclairée pour l'inciter à s'y asseoir. Une fois la personne assise la luminosité passe à 10% dans la pièce et la lampe montrant le fauteuil berçant s'éteint. Après un temps estimé suffisant pour sa relaxation, des indicateurs lumineux sont affichés sur la télévision et une icône rappelant le sommeil s'illumine pour inciter

3.4. EXEMPLES DE SCÉNARIOS



Figure 3.3 – Utilisation du varioglass



Figure 3.4 – Exemple d'indicateur lumineux

la personne à retourner se coucher (voir figure 3.4). Quelques instants plus tard, la musique s'arrête, le chemin lumineux vers la chambre s'allume et l'éclairage passe à 10%. Lorsqu'elle quitte le salon, le téléviseur s'éteint. Une fois la personne détectée dans le corridor, la lumière de la chambre s'allume à 25% et la lumière du salon s'éteint. De retour dans la chambre, le chemin lumineux s'éteint et la lumière de la chambre passe à 10%. Lorsqu'elle retourne dans son lit, la lumière de la chambre s'éteint.

3.5. MODÉLISATION

3.4.4 Scénario 4 : Errance nocturne

On parle d'errance nocturne lorsqu'une personne (souffrant de la maladie d'Alzheimer) se réveille dans la nuit pour satisfaire un ou plusieurs besoins, mais s'engage dans la réalisation d'une autre activité ou d'une errance sans but précis. L'agitation et l'angoisse lui font oublier de retourner se coucher, une confusion s'installe et s'empare d'elle.

Pour construire ce scénario d'errance nocturne, nous nous servons des trois scénarios précédents. En effet, le scénario d'errance nocturne consiste à satisfaire un besoin quelconque (boire un verre d'eau ou aller aux toilettes) puis offrir dans le salon une atmosphère de confort, de relaxation pour calmer une quelconque angoisse ou agitation avant de raccompagner la personne dans sa chambre.

Proposition

Errance nocturne : C'est toute combinaison du scénario 1 ou scénario 2 répétée une ou plusieurs fois suivies du scénario 3.

3.5 Modélisation

En nous basant sur le modèle HTA d'analyse des tâches, nous développons dans cette section la modélisation retenue. Initialement, le modèle HTA a été développé pour pallier aux limitations des méthodes d'analyse classiques basées sur le temps et le mouvement pour l'analyse des tâches complexes et non répétitives [105]. Par ailleurs, il permet une adaptation facile à la gestion des AVQ. Nous nous sommes aussi inspirés du modèle CTT pour la gestion des opérateurs et certaines décompositions.

Le modèle proposé est un quintuple $\Gamma = (X, O, A, R, \vartheta)$ où

X est l'ensemble fini de nœuds de l'arbre.

O l'ensemble fini des opérateurs.

A la liste des actions

$X \setminus A$ est l'ensemble fini des nœuds internes.

R une application ternaire entre un nœud, un opérateur et ses fils définit par :

$\forall \eta \in X \setminus A, \alpha \in O, (\eta, \alpha, \Gamma(\eta))$ où $\Gamma(\eta)$ est le sous-arbre de profondeur 1 avec comme

3.5. MODÉLISATION

racine η .

$\vartheta \in X$ est le nœud racine.

Ce modèle hiérarchique de représentation des scénarios se définit comme suit.

1. Un scénario est un arbre ayant un but (nœud racine), plusieurs tâches (nœuds internes) et plusieurs actions représentées par des nœuds terminaux ;
2. Le but est l'état de satisfaction à atteindre ;
3. Les tâches et les sous-tâches sont la décomposition utilisée pour satisfaire le but ;
4. Les décompositions de tâches qui donnent une seule sous-tâche ne sont pas permises ;
5. Les sous-tâches d'une tâche sont considérées comme frères et sont liées par un opérateur ;
6. L'opérateur n-aire est porté par le parent et s'applique à tous les nœuds enfants.
7. L'opérateur u-naire s'applique directement au nœud qui le porte.
8. Des préconditions et des posts conditions peuvent être associées à des tâches ;
9. Une tâche peut être effectuée par un ou plusieurs utilisateurs ;
10. La décomposition se termine lorsqu'une sous-tâche est considérée comme tâche élémentaire ;
11. Une tâche élémentaire est considérée comme une action ;
12. Une action est équivalente à une action physique et informatique compréhensible par des programmes, ou associée à un ensemble de capteurs ou d'effecteurs ;
13. Les actions décrivent les interactions entre l'utilisateur, les objets, l'environnement et le système.

Les opérateurs sont de plusieurs types, la liste complète de ceux-ci peut être trouvée dans le document CTT [77]. Pour des besoins de simplification, nous allons retenir uniquement les opérateurs suivants :

3.5. MODÉLISATION

L'opérateur «Choice» : c'est un opérateur binaire. Il traduit le "ou" inclusif. Cependant, dans le modèle proposé, une fois que l'un des opérandes de l'opérateur est démarré, l'autre est désactivé même s'il n'est pas terminé. Pour illustrer ce concept considérons l'activité boire un verre d'eau. Cette activité liée via l'opérateur de choix les deux sous-tâches suivantes :

- Boire un verre d'eau au robinet ;
- Boire un verre d'eau depuis un contenant au réfrigérateur.

L'opérateur «Sequential» : c'est un opérateur binaire qui indique que les opérandes s'exécutent de manière séquentielle, c'est-à-dire l'un à la suite de l'autre. Par exemple "aller à la salle de bain depuis la chambre" implique de réaliser séquentiellement les sous-tâches suivantes

- Sortir de la chambre ;
- Marcher vers la salle de bain ;
- Entrer dans la salle de bain.

L'opérateur «Concurrent» : c'est un opérateur qui indique que les tâches se déroulent de manière concurrente. Les tâches sont effectuées dans n'importe quel ordre, c'est-à-dire en même temps, y compris la possibilité de démarrer une tâche avant que l'autre ne soit terminée. C'est un opérateur binaire. Par exemple lors de l'exécution du scénario "préparer du thé en écoutant les informations à la radio" les tâches suivantes peuvent se faire en parallèle :

- Faire bouillir de l'eau dans la théière ;
- Écouter la radio ;
- Mettre le thé dans la tasse.

L'opérateur «Disabling» : cet opérateur n-aire permet d'interrompre complètement la tâche en cours pour activer la tâche suivante. Un exemple durant l'activité "aller se coucher" pourrait être d'arrêter le téléviseur si les sous-tâches suivantes sont impliquées dans l'activité :

3.5. MODÉLISATION

- Écouter la télévision ;
- Sortir du salon et se diriger vers la chambre.

L'opérateur «Independence» : c'est un opérateur qui indique que les tâches peuvent être effectuées dans n'importe quel ordre. Toutefois, une tâche commencée doit être terminée avant de passer à la suivante. La notion d'ordre n'a pas d'importance. Durant l'activité "faire une tasse de café au lait" les sous-tâches suivantes sont effectuées de manière indépendante :

- Prendre une tasse ;
- Prendre la boîte de lait ;
- Prendre une petite cuillère ;
- Prendre du sucre ;
- Prendre la boîte à café.

L'opérateur «Exclusive Choice» : c'est un opérateur qui indique le choix. Il s'agit d'un choix exclusif au sens mathématique du terme entre les différentes tâches. Pour l'activité "dormir" on pourrait soit réaliser exclusivement l'une des sous-tâches suivantes :

- Se coucher sur le lit ;
- Se coucher sur la chaise berçante.

L'opérateur «Interweaving» : c'est un opérateur d'entrelacement utilisé pour indiquer que des tâches peuvent s'exécuter de manière entrelacée. Par exemple durant l'activité "boire un verre d'eau au robinet" on pourrait réaliser de manière croisée les sous-tâches suivantes :

- Ouvrir le placard pour prendre un verre
- Ouvrir le robinet pour laisser couler l'eau

3.5. MODÉLISATION

L'opérateur «Optional» : c'est un opérateur u-naire qui indique que la tâche peut être optionnelle. Dans l'activité "écouter la télévision au salon" les sous-tâches suivantes sont faites de manière optionnelle :

- S'asseoir sur le divan pour écouter la télévision ;
- Rester debout pour écouter la télévision.

L'opérateur «Repeat» : c'est un opérateur u-naire qui indique que la tâche peut être effectuée plusieurs fois. L'activité "boire un verre d'eau" peut se faire plusieurs fois dans le même scénario avant de passer à l'étape suivante.

Le processus de création de scénarios est un processus itératif et incrémental permettant de concevoir des approches qui non seulement maintiennent les parties prenantes motivées et engagées, mais leur fournissent également une expérience de conception qui améliore efficacement leurs compétences et maximise leur objectif. Trois types d'approches de conception ont été identifiés : approche ascendante, approche descendante, approche par combinaison.

3.5.1 Approche ascendante (Bottom-up Approach)

Cette approche utilise l'expérience de la personne pour construire un scénario qui correspond à ses besoins. Partant des actions effectuées pour la réalisation des AVQ, on construit des abstractions hiérarchiques correspondant à un regroupement de plusieurs activités. De ces abstractions, on crée d'autres abstractions, ainsi de suite jusqu'à parvenir au but à atteindre. Cette approche correspondrait mieux à celle adoptée dans le paradigme DIY. En effet, les personnes qui souhaitent créer des scénarios par elles-mêmes vont s'inspirer des capteurs déjà disponibles pour créer une nouvelle façon de faire des choses. Le besoin dans ce cas émerge à partir des dispositifs et du matériel disponible. Dans cette situation, les objets du quotidien et les équipements électroniques sont souvent modifiés pour prendre en compte les exigences du quotidien. La Figure 3.5 montre un exemple de construction de scénarios de manière ascendante ou via les données des capteurs.

Cette approche est centrée utilisateur, car elle commence par des habitudes de la personne pour bâtir un scénario correspondant à ses besoins. La question principale

3.5. MODÉLISATION

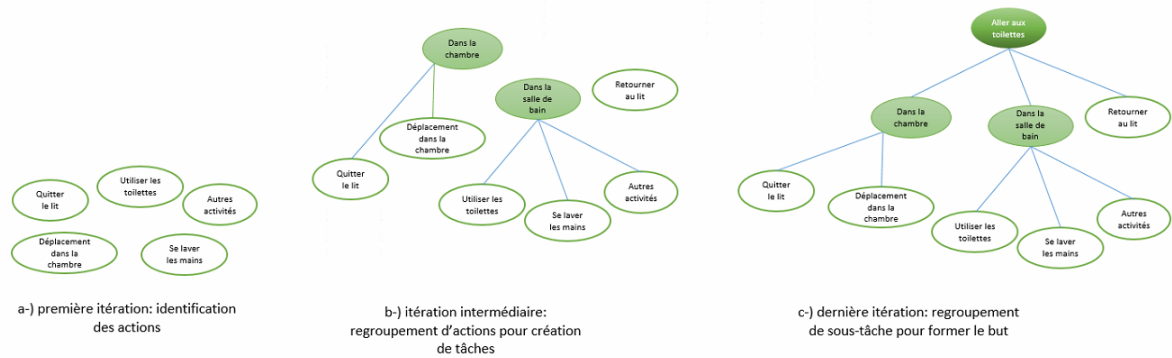


Figure 3.5 – Approche ascendante de création de scénarios

que l'on se pose est : **qu'est-ce que l'utilisateur fait ?**

3.5.2 Approche descendante (Top Down Approach)

Dans cette approche, la question principale est de savoir comment la technologie pourrait être utilisée pour faciliter la réalisation des AVQ. Le traitement et l'analyse de cette question imposent donc de partir de l'objectif à atteindre pour se rendre aux outils technologies disponibles. Ainsi partant du but principal et par raffinements successifs le but est décomposé en tâches de plus en plus simples jusqu'à atteindre un état atomique qui est l'action que les capteurs sont capables de détecter ou de superviser. Les concepteurs modélisent la hiérarchie, la profondeur de l'arbre, les différentes décompositions, la planification des tâches et la priorisation des besoins qui sont pris en compte. La Figure 3.6 illustre ce processus de création en commençant par le but pour déboucher aux actions.

La question posée ici est : **qu'est-ce que l'utilisateur peut faire ?**

3.5.3 Approche par combinaison

Dans cette approche on se propose d'utiliser la totalité ou un sous-ensemble d'un scénario existant pour bâtir un nouveau scénario ou en modifier un. Nous avons identifié deux combinaisons possibles : Création d'un nouveau scénario en utilisant des scénarios existants et Combinaison sans création d'un nouveau scénario.

3.5. MODÉLISATION

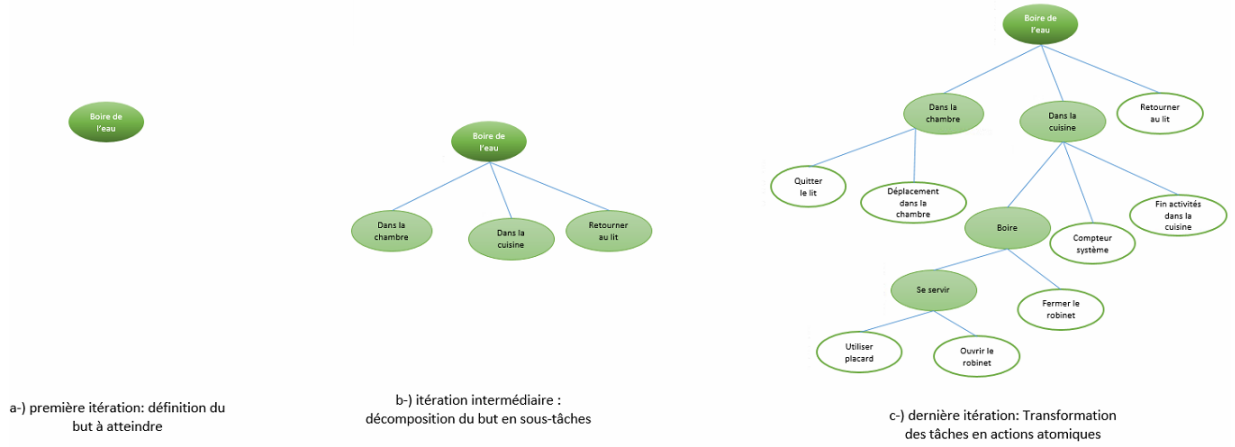


Figure 3.6 – Approche descendante pour la création de scénarios

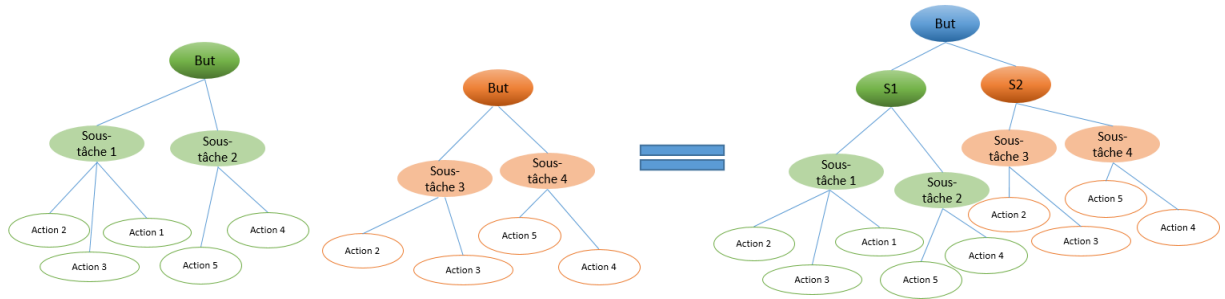


Figure 3.7 – Combinaison de scénarios : création d'un nouveau scénario

Création d'un nouveau scénario en utilisant des scénarios existants

Soit deux scénarios S_1 et S_2 , le scénario résultat S est tel que sa racine soit parent des racines de S_1 et S_2 .

En d'autres termes

le nouveau scénario est donné par $S = (X, O, A, R, \vartheta)$

où $S_1 = (X_1, O_1, A_1, R_1, \vartheta_1)$ et $S_2 = (X_2, O_2, A_2, R_2, \vartheta_2)$

avec $R = (\vartheta, \alpha, (\vartheta_1 \cup \vartheta_2))$ la nouvelle relation qui lie S_1 et S_2 à S via l'opérateur α .

Une illustration de cette combinaison est donnée à la Figure 3.7.

3.6. FORMALISME

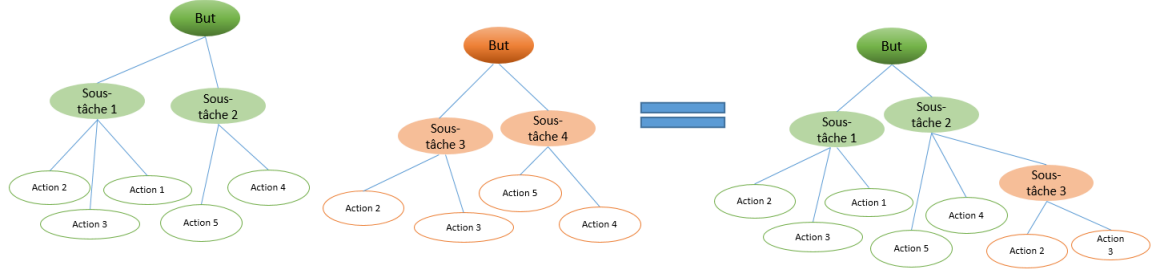


Figure 3.8 – Combinaison sans création d’un nouveau scénario

Combinaison sans création d’un nouveau scénario

Soit deux scénarios S_1 et S_2 le scénario résultant S est une copie du scénario S_1 dans lequel est injecté une partie ou la totalité du scénario S_2 à la position x_1 .

Formellement, $S = (X, O, A, R, \vartheta)$

avec $S_1 = (X_1, O_1, A_1, R_1, \vartheta_1)$ et $S_2 = (X_2, O_2, A_2, R_2, \vartheta_2)$

où $R = (x_1, \alpha, (S_1(x_1) \cup S_2(x_2)))$, $S(x)$ représente le sous arbre ayant pour racine le noeud x . Une illustration de cette combinaison est donnée à la Figure 3.8.

Pour simplifier notre implémentation, nous avons décidé que les nouveaux scénarios sont des copies plutôt que des références vers les noeuds. Ce choix a été fait pour limiter la complexité à l’implémentation et faciliter la recherche de scénarios à exécuter.

3.6 Formalisme

Quand le problème est identifié, la spécification permet d’apporter une solution. Étant donné une application d’assistance à réaliser, quelle démarche mettre en place pour spécifier le modèle de scénario correspondant ? Comment aider l’utilisateur à bâtir un scénario ? Comment s’assurer que le scénario bâti est conforme à la spécification ? Telles sont les questions posées dans ce paragraphe.

Il est impossible de décrire l’ensemble des situations possibles rencontrées par l’utilisateur final durant la phase de conception. Il faut donc mettre en place une spécification globale et s’assurer à tout instant durant la phase de conception que les ajouts itératifs n’entrent pas en conflit avec la spécification de base. Chaque spé-

3.6. FORMALISME

cification est construite autour d'axiomes fournissant des propriétés pour décrire le comportement du système d'assistance. Itérativement et de manière incrémentale, chaque axiome est ajouté au modèle, traduit en un système de preuve et passé à un moteur de raisonnement.

3.6.1 Le choix de ALLOY comme outil de spécifications formelles

Les spécifications formelles sont le plus souvent utilisées pour concevoir, valider, documenter, communiquer, réorganiser et réutiliser des solutions [61]. Plusieurs outils de spécifications et de vérifications formelles ont été développés dans la littérature. Entre autres, nous pouvons citer, les méthodes B et ses dérivées, le langage OCL, le langage VDM, le langage Z, ou le langage ALLOY. Nous utilisons ALLOY comme outil de spécification et de validation de nos scénarios. En effet, ALLOY est un langage qui capture l'essence des abstractions logicielles de manière simple et succincte. Il effectue une analyse entièrement automatique et peut exposer les fautes logicielles les plus subtiles. ALLOY a été choisi parce qu'il permet de produire un modèle abstrait du système à spécifier. Ceci facilite son évolution ou son expansion pour le futur et pour les éventuelles modifications. Ainsi, ALLOY est capable de garantir que les nouvelles modifications d'un système sont compatibles avec ses spécifications d'origine. Ce choix est soutenu par le fait que dans le domaine des scénarios les évolutions sont courantes. Enfin, ALLOY peut être utilisé avant, pendant et durant la phase de conception et même pendant celle de mise en oeuvre.

3.6.2 Du Scénario à la spécification formelle en ALLOY

Nous avons combiné à l'approche dirigée par les scénarios un langage de logique orienté objet pour formaliser la définition d'un scénario. L'approche utilisée garantit par validation que chaque instance d'un nouveau scénario est conforme à la spécification de base.

Une spécification ALLOY est basée sur des signatures. Une signature est similaire à la notion de classe en programmation orientée objet. Cependant, les signatures ne contiennent pas de méthodes. Une spécification ALLOY est avant tout constituée d'un

3.6. FORMALISME

```

module scenarios

sig User{}
sig Room{}
sig Furniture{}
sig Operator{}
sig Condition{}

enum TaskType {abstractTask, interactionTask, systemTask, userTask}

abstract sig Task{
  taskType : TaskType,
  user : set User,
  parent : lone CompositeTask,
  precondition : some Condition,
  postcondition : some Condition
}

sig CompositeTask extends Task{
  operator : one Operator
}

sig Action extends Task{
  room : one Room,
  furniture : set Furniture
}

one sig Goal extends CompositeTask {} {
  no parent
}

fact {
  // All tasks are reachable from Goal
  Task in Goal.*~parent
  // Action are not abstract
  not (abstractTask in Action.taskType)
  //No decomposition into one element is permitted
  all s: CompositeTask | #(s.{~parent}) > 1 or #(s.{~parent}) = 0
  //The goal is always an abstract type
  Goal.taskType = abstractTask
  //All tasks except Goal have a parent
  all s: Task-Goal | one s.parent
}

assert acyclic {
  // No cycle
  no s: Task | s in s.^{~parent}
}

```

Figure 3.9 – Exemple de spécification de base d’un scénario en ALLOY

ensemble de faits et d’assertions. La Figure 3.9 montre une vue partielle contenant les assertions et les faits pour la description du modèle décrit à la section 3.5.

Les signatures USER, ROOM, FURNITURE, OPERATOR et CONDITION n’ont pas de définitions spécifiques, elles sont utilisées pour exprimer les relations avec les autres signatures. La classe TASK est définie comme étant une signature abstraite spécialisée par les signatures COMPOSITETASK et ACTION. Le mot-clé *enum* permet de définir une énumération des différentes tâches. La signature GOAL qui spécialise la signature COMPOSITETASK utilise le mot-clé *one* pour indiquer qu’il ne peut contenir qu’un seul objet. La signature User contient les objets utilisateurs impliqués dans le scénario. Le mot-clé *lone* associé à l’attribut parent indique que chaque tâche a au plus un parent. La signature GOAL n’a pas de parent comme l’indique l’attribut no parent. Dans ALLOY, les attributs sont représentés par des relations binaires. Par exemple, l’attribut parent est un sous-ensemble du produit cartésien TASK*COMPOSITETASK.

3.6.3 La spécification de contraintes

ALLOY propose deux manières pour exprimer les contraintes. La première est l’utilisation des contraintes sur les multiplicités d’ensemble ou sur des relations entre les signatures. La seconde est l’utilisation des faits énonçant une contrainte sur l’en-

3.6. FORMALISME

semble des relations.

Les faits introduits par le mot-clé *fact*, sont des contraintes qui doivent être satisfaites par une instance de la spécification, c'est-à-dire un ensemble de valeurs pour les signatures et leurs attributs. Plusieurs faits ont été définis pour exprimer les contraintes à respecter pour les scénarios comme le montre la Figure 3.9.

Fait 1 - l'atteignabilité exprime le fait que toutes les tâches sont accessibles depuis la racine du scénario. Il s'agit de fixer une contrainte pour dire que le but est atteignable depuis une tâche quelconque. Nous avons utilisé la composition de deux relations et un opérateur de fermeture pour exprimer cette contrainte.

Fait 2 – la non décomposition des actions exprime le fait que les actions ne sont pas des tâches abstraites. Par conséquent, elles ne peuvent pas être décomposées en sous-tâches.

Fait 3 – la décomposition des tâches traduit le fait que toutes les décompositions doivent se terminer par des actions. Il s'agit de s'assurer que chaque parent a au moins deux enfants ou aucun enfant.

Fait 4 – la généralité du but s'exprime par le fait que le but est toujours une tâche abstraite et peut ne pas se décomposer en plusieurs tâches.

Fait 5 – l'unicité du parent impose à toutes les tâches, sauf le but de n'avoir qu'un seul parent direct.

Fait 6 – les opérateurs n-aire sont définis sur des nœuds composites et s'appliquent aux nœuds enfants.

Fait 7 – les opérateurs u-naïres peuvent être définis sur tous les nœuds.

Fait 8 – l'unicité de l'opérateur u-naïres s'applique sur chaque nœuds.

3.6. FORMALISME

Validation des faits

La validation des faits est un processus qui recherche des instances de la spécification pour lesquelles l'une des instances violerait un fait. Dans ALLOY ce processus se fait en utilisant la commande *run n*. Cette commande recherche l'existence d'une instance de la spécification avec au plus *n* objets dans chaque signature qui viole une contrainte. Le nombre *n* représente la portée (l'étendue) dans laquelle nous effectuons notre recherche. La qualité du résultat dépend du choix de cette portée.

Pour comprendre la notion de portée, considérons la commande *run 1*. Elle permet de vérifier que les instances de scénarios aléatoires n'ayant qu'une seule signature et ne violant aucune contrainte sont possibles. Dans le cas d'espèce le seul scénario possible est celui ayant 1 nœud : le but. Puisque, le but ne viole aucune des contraintes alors cette validation est correcte dans cette portée de 1. Un second exemple est celui donné par la commande *run for 1 but exactly 3 TASK*, qui permet de trouver des spécifications pour lesquelles on aurait un objet pour chacune des signatures à l'exception de la signature TASK qui en aurait 3. Cette validation elle aussi est correcte, car il existe bien un scénario avec 3 tâches où chaque parent a plus d'un enfant. En revanche la commande *run for 1 but exactly 2 TASK* échoue, puisqu'il est impossible de construire d'instance de scénarios ayant 2 tâches où chaque parent doit avoir plus d'un enfant.

La vérification des assertions

Une assertion est un théorème qui utilise les faits pour prouver que les nouvelles prépositions et spécifications ne sont pas contradictoires à la spécification de base. ALLOY prouve les assertions en analysant toutes les instances possibles d'une spécification dans une portée bien précise. Une assertion est une contrainte destinée à être valide pour tous les cas possibles dans une portée (l'étendue, espace, limite) bien définie. Pour vérifier les assertions d'une spécification, nous utilisons ALLOY pour lui demander de trouver des contre-exemples pour lesquels le théorème ne serait pas vrai. Un exemple d'assertion est le fait de dire qu'il ne devrait pas y avoir de cycle dans les relations entre parents et enfants.

ALLOY utilise la commande *check n* où *n* représente la portée dans laquelle la

3.7. IMPLÉMENTATION D'UNE INSTANCE DE LA SPÉCIFICATION : CAS DU SCÉNARIO D'ERRANCE NOCTURNE

```
one sig NighttimeWandering extends Goal{}

fact NighttimeWandering_Scenario{
}

run NighttimeWandering_Scenario {} for 10
```

Figure 3.10 – Exemple de spécification de base d'un scénario en ALLOY

vérification est faite. ALLOY vérifie que toutes les instances de la spécification dans cette plage ne violent aucun fait. Si aucun contre-exemple n'est trouvé, cela ne garantit pas le résultat, mais plutôt qu'aucun contre-exemple n'a été trouvé dans cette portée. Généralement, ALLOY traduit les assertions en formules booléennes et utilise un solveur basé sur SAT (SATISFIABILITY) pour vérifier chaque formule.

3.7 Implémentation d'une instance de la spécification : cas du scénario d'errance nocturne

Cette section présente l'implémentation en ALLOY du scénario d'errance nocturne. Elle décrit de manière itérative comment la représentation hiérarchique du scénario se traduit en langage formel, puis, indique la démarche de vérification et de validation automatique qui a été faite.

A la création du nœud but, racine du scénario, un fichier de spécification est créé. Initialement, le modèle contient uniquement le code présenté à la Figure 3.10 qui est lié à la spécification de base définie à la Figure 3.9. Ce code indique que le nœud racine du scénario est un but. Puis la commande «`RUN NIGHTTIMEWANDERING_SCENARIO {} FOR 10`» va rechercher les instances dans une portée de 10, pour lesquelles, les faits de la spécification de base sont vrais. A ce stade de la spécification, seul le fait de base « le but est une tâche » est respecté. Le code générée est donc conforme à la spécification de base et la Figure C.1 de l'Annexe C montre les instances obtenues.

À l'ajout d'une tâche ou d'une action dans le scénario graphique (pour l'instant un fichier), ALLOY ajoute automatiquement à la spécification «`NIGHTTIMEWANDERING_SCENARIO`» les signatures correspondantes. Chaque modification du scénario

3.7. IMPLÉMENTATION D'UNE INSTANCE DE LA SPÉCIFICATION : CAS DU SCÉNARIO D'ERRANCE NOCTURNE

graphique met à jour la spécification. La Figure 3.11 montre les tâches composites créées, les opérateurs, les tâches parents et les relations entre les nœuds. Dans le cas du scénario, les signatures des tâches composites suivantes sont générées « ASSISTANCE », « INCENTIVEUSER », « BACKINROOM », etc. Elles sont créées comme étant des sous-classes de la signature « COMPOSITETASK ». Puis les relations avec la classe parente « NIGHTTIMEWANDERING » sont construites par les instructions « GETOUTROOM.PARENT = NIGHTTIMEWANDERING », « ASSISTANCE.PARENT = NIGHTTIMEWANDERING » et « INCENTIVEUSER.PARENT = NIGHTTIMEWANDERING ». Désormais, les tâches « GETOUTROOM », « ASSISTANCE » et « INCENTIVEUSER » ont pour parent la tâche « NIGHTTIMEWANDERING ».

Par la suite, à l'association de la position hiérarchique de chaque nœud dans le scénario, la spécification se met à jour via la commande suivante :

« NIGHTTIMEWANDERING.(~ PARENT) = GETOUTROOM + ASSISTANCE + INCENTIVEUSER + BACKINROOM ».

Les opérateurs sont automatiquement stockés dans les nœuds correspondants, l'instruction « GETOUTROOM.OPERATOR = TRANS_OP » permet d'ajouter l'opérateur « TRANS_OP » à la tâche « GETOUTROOM ». Toutes ces informations sont encapsulées dans un fait, pour le cas d'espèce, le fait a le nom « NIGHTTIMEWANDERING_SCENARIO » (voir Figure 3.11).

La validation automatique avec la même commande à chaque itération montre qu'il n'y a pas d'erreurs et qu'une instance a été trouvée comme le montre le graphique de la Figure C.2 de l'annexe C. Ainsi de manière itérative et incrémentale les actions sont ajoutés, et à chaque modification la commande « RUN NIGHTTIMEWANDERING_SCENARIO {} FOR 10 » est appelée pour trouver une instance. Si aucun fait n'est violé l'exécution se poursuit. Puis à la fin du processus de modélisation la commande « CHECK ACYCLIC FOR P » avec $p = \{3, 5, 10, 15\}$ est exécutée pour trouver des contre-exemples. Les résultats de cette vérification d'assertions sont donnés dans les paragraphes suivants.

3.7. IMPLÉMENTATION D'UNE INSTANCE DE LA SPÉCIFICATION : CAS DU SCÉNARIO D'ERRANCE NOCTURNE

```
one sig NighttimeWandering extends Goal{}

one sig GetOutRoom,
  Assistance, IncentiveUser, BackInRoom,
  SatisfyNeed, NoNeed, AssistPerson,
  NeedIdentification, Toilet, Salon,
  Kitchen extends CompositeTask{}

one sig add_op, choice_op,
  choice_trans_op, trans_op,
  inter_op extends Operator {}

fact NighttimeWandering_Scenario{

  GetOutRoom.parent = NighttimeWandering
  Assistance.parent = NighttimeWandering
  IncentiveUser.parent = NighttimeWandering
  BackInRoom.parent = NighttimeWandering
  SatisfyNeed.parent = Assistance
  NoNeed.parent = Assistance
  AssistPerson.parent = SatisfyNeed
  NeedIdentification.parent = SatisfyNeed
  Toilet.parent = NeedIdentification
  Salon.parent = NeedIdentification
  Kitchen.parent = NeedIdentification

  NighttimeWandering.(~parent) = GetOutRoom + Assistance
  + IncentiveUser + BackInRoom
  Assistance.(~parent) = SatisfyNeed + NoNeed
  SatisfyNeed.(~parent) = NeedIdentification + AssistPerson
  NeedIdentification.(~parent) = Toilet + Salon + Kitchen

  GetOutRoom.operator = trans_op
}

run NighttimeWandering_Scenario {} for 10
```

Figure 3.11 – Exemple de spécification de base d'un scénario en ALLOY

3.8. RÉSULTATS DE LA VALIDATION ET DE LA SPÉCIFICATION DE SCÉNARIOS

Tableau 3.1 – Résultats de l’exécution de l’assertion de la spécification de base

	Assertion acyclique		
Portée	Variables	Clauses	Temps (ms)
3	713	1083	18
5	2212	19988	41
10	10781	19988	41
15	28545	54266	1226549

3.8 Résultats de la validation et de la spécification de scénarios

Nous avons soumis à ALLOY pour analyse, la spécification de base tel que donnée à la Figure 3.9. Par la suite, nous nous sommes attelés à créer des instances pour nos trois différents scénarios pour savoir si l’une d’elles n’était pas conforme à la spécification de base.

Le Tableau 3.1 montre les résultats obtenus en exécutant l’assertion acyclique de la spécification de base. On peut observer que pour la portée 10, 10 781 variables ont été utilisées pour générer 19 988 clauses en 1.8 minute et qu’aucun contre-exemple n’a été trouvé. Nous pouvons donc conclure que dans cette portée la spécification reste toujours valide.

Les résultats d’exécution de trois scénarios (voir section 3.4) conçus et mis en place au laboratoire DOMUS et qui ont été expérimentés durant plusieurs mois au cours de deux expérimentations à domicile chez des personnes âgées sont présentés ci-dessous.

3.8.1 Résultats de la spécification du scénario 1

Le scénario « aller à la salle de bain durant la nuit depuis la chambre à coucher » décrit le comportement d’une personne durant la nuit à qui on souhaiterait lui porter assistance pour aller satisfaire un besoin à la salle de bain.

Deux principales tâches ont été identifiées pour la réalisation de ce scénario :

1. Dans la chambre ;
2. Dans la salle de bain ;

3.8. RÉSULTATS DE LA VALIDATION ET DE LA SPÉCIFICATION DE SCÉNARIOS

Tableau 3.2 – Résultats de la vérification des assertions pour le scénario 1.

Portée	Assertion acyclique		
	Variables	Clauses	Temps (ms)
3	48 236	80 978	47
4	48 530	81 386	125
5	48 824	81 794	47
10	51 051	85 362	141
15	53 341	88 812	125

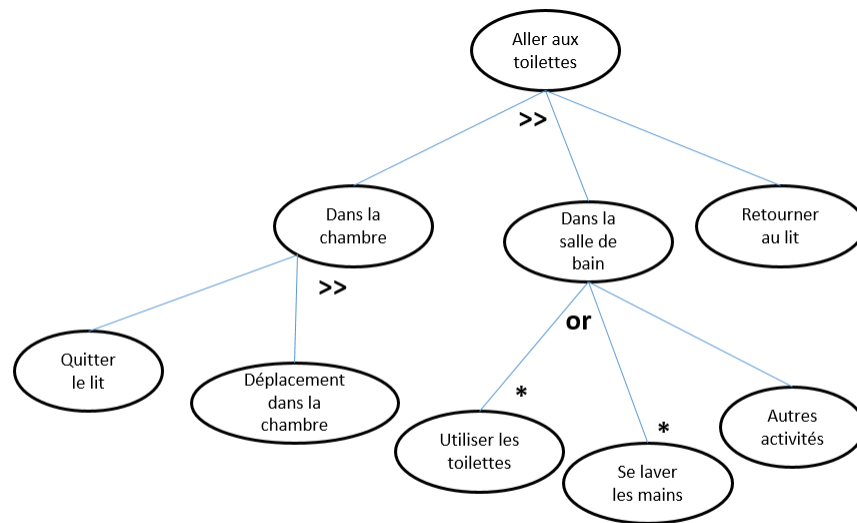
Sept actions ont été identifiées :

1. Retourner au lit ;
2. Quitter le lit ;
3. Déplacement dans la chambre ;
4. Utiliser les toilettes ;
5. Se laver les mains ;
6. Autres activités ;

La Figure 3.12 montre la représentation hiérarchique obtenue via le processus de modélisation dirigé par les scénarios que nous avons utilisé. Comme le montre la figure 3.12, l'activité commence dans la chambre et se termine dans celle-ci. Les sous-tâches de la tâche utiliser les toilettes, se laver les mains et autres activités sont des activités non obligatoires. Elles peuvent toutes, ou au moins une, être exécutées. C'est la raison pour laquelle elles sont liées avec l'opérateur « or ». La tâche « autres activités » peut tout aussi bien représenter un scénario complet ou une simple tâche ou action. L'opérateur « * » utilisé sur la tâche « utiliser les toilettes » marque le fait que cette tâche peut être répétitive.

Le Tableau 3.2 décrit les résultats de la spécification produite pour le scénario « aller à la salle de bain durant la nuit depuis la chambre à coucher ». Il présente la vérification de l'assertion acyclique décrite dans la spécification de base voir (Figure 3.12). Il en ressort que pour une portée de 10 avec 51051 variables de générées pour former 85 362 clauses aucun contre-exemple n'a été trouvé en 141 ms.

3.8. RÉSULTATS DE LA VALIDATION ET DE LA SPÉCIFICATION DE SCÉNARIOS



>> : séquentielle, or : choix, [] : optionnelle, ||| : entrelacement, || : parallèle, * : répétition

Figure 3.12 – Représentation hiérarchique du scénario : aller aux toilettes

3.8.2 Résultats de la spécification du scénario 2

Le scénario « boire un verre d’eau au robinet de la cuisine » décrit le comportement d’une personne à qui on souhaiterait apporter une assistance pour aller satisfaire un besoin durant la nuit.

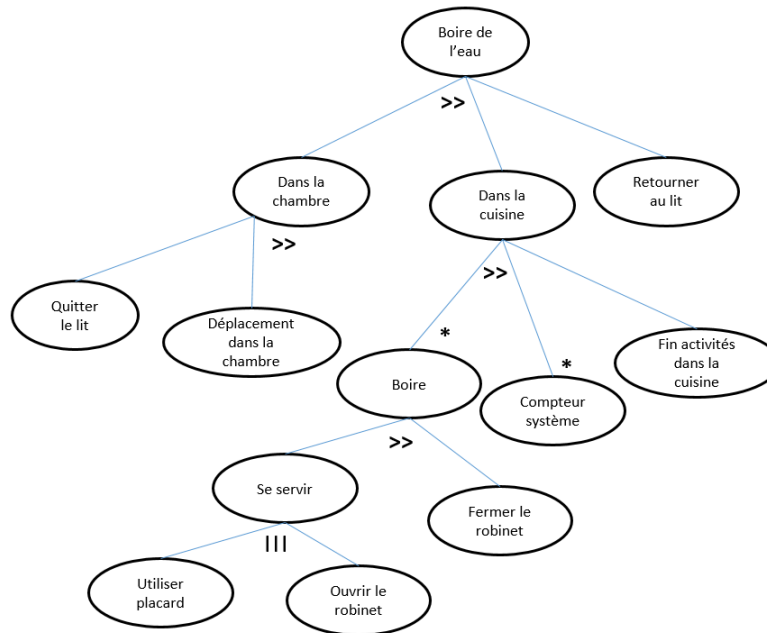
La conception de ce scénario permet de dégager quatre principales tâches nécessaires à sa réalisation. Ces tâches sont :

1. Dans la chambre ;
2. Dans la cuisine ;
3. Boire ;
4. Se servir ;

huit actions ont été identifiées :

1. Quitter le lit ;
2. Déplacement dans la chambre ;
3. Compteur système ;
4. Fin activités dans la cuisine ;

3.8. RÉSULTATS DE LA VALIDATION ET DE LA SPÉCIFICATION DE SCÉNARIOS



>> : séquentielle, or : choix, [] : optionnelle, ||| : entrelacement, || : parallèle, * : répétition

Figure 3.13 – Représentation hiérarchique du scénario : «boire de l'eau à la cuisine»

5. Fermer le robinet ;
6. Utiliser placard ;
7. Ouvrir le robinet ;
8. Retourner au lit ;

La représentation hiérarchique de la Figure 3.13 montre que les tâches suivantes « dans la chambre », « dans la cuisine » et « retourne au lit » sont séquentielles et sont exécutées dans cet ordre. Pour matérialiser le fait que la personne peut boire plusieurs fois nous avons utilisé l'opérateur u-naire « * ». L'entrelacement des tâches d'utilisation du placard et d'ouverture du robinet est indiqué par le symbole « ||| ». La tâche « compteur système » est une estimation du temps qu'il faudrait à la personne pour boire son verre d'eau.

Le Tableau 3.3 décrit les résultats obtenus pour la vérification des assertions de l'instance des spécifications pour le scénario 2. La même lecture que le tableau précédent est faite ici.

3.8. RÉSULTATS DE LA VALIDATION ET DE LA SPÉCIFICATION DE SCÉNARIOS

Tableau 3.3 – Résultats de la vérification des assertions pour le scénario 2

	Assertion acyclique		
Portée	Variables	Clauses	Temps (ms)
3	4 804	7 870	47
4	4 980	8 121	110
5	5 156	8 372	94
10	8 229	13 296	63
15	24 798	40 386	156

Tableau 3.4 – Résultats de la vérification des assertions pour le scénario 3

	Assertion acyclique		
Portée	Variables	Clauses	Temps (ms)
3	14 061	26 231	56
4	14 291	26 563	55
5	14 521	26 895	43
10	17 584	33 579	57
15	35 330	69 051	116

3.8.3 Résultats de la spécification du scénario 3

Le scénario « Se relaxer au salon » décrit le comportement d’une personne durant la nuit à qui on souhaiterait porter une assistance pour calmer son anxiété.

La conception de ce scénario présuppose que l’anxiété et l’agitation de la personne se produisent une fois qu’elle est dans la chambre. La Figure 3.14 montre hiérarchiquement comment les différentes tâches s’emboîtent pour offrir une ambiance de détente à la personne. La tâche «d’installation sur le fauteuil» est optionnelle en ce sens que c’est le profil de la personne qui guide le choix pour proposer une façon de se relaxer. Le choix de la musique ou des écrans de veille relaxants est spécifié par le profil de la personne. La fin de la relaxation s’accompagne des messages d’incitation à retourner se coucher.

Le Tableau 3.4 décrit les résultats obtenus pour la vérification des assertions de l’instance de la spécification. Aucun contre-exemple n’a été trouvé dans une portée de 10. Le scénario est donc vérifié pour cette portée.

3.9. CONCLUSION DU CHAPITRE

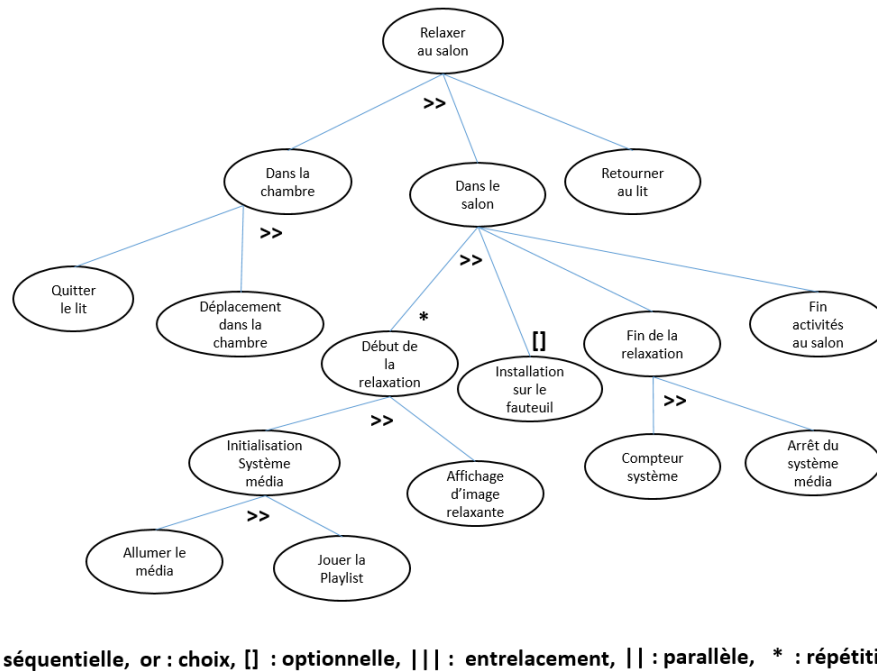


Figure 3.14 – Représentation hiérarchique du scénario : «se relaxer au salon»

3.8.4 Résultats de la spécification du scénario 4

Le scénario « errance nocturne » décrit le comportement d'une personne durant la nuit à qui on souhaiterait porter assistance durant des épisodes d'errances nocturnes pour en réduire l'impact sur elle et sur ses proches.

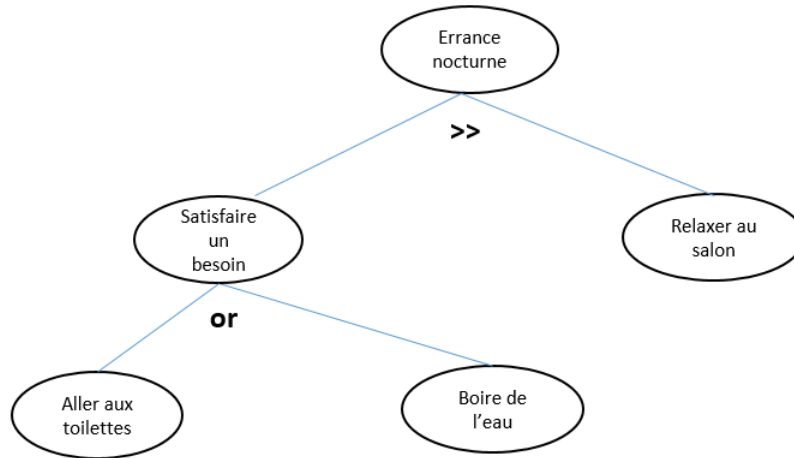
La démarche de conception nous permet de réduire le scénario d'errance nocturne à sa plus simple expression. La Figure 3.15 montre une représentation hiérarchique dans laquelle l'attention est tout de suite portée sur l'utilisateur. Elle décline le scénario en 2 axes majeurs : «satisfaire un besoin» et «relaxer la personne».

Le Tableau 3.5 décrit les résultats obtenus pour la vérification des spécifications.

3.9 Conclusion du chapitre

Dans un contexte d'habitat intelligent, la mécanique sous-jacente d'assistance doit être telle que la sûreté de fonctionnement doit être garantie et la sécurité des personnes assurée d'un point de vue des besoins utilisateurs. La mécanique de test des logiciels

3.9. CONCLUSION DU CHAPITRE



>> : séquentielle, or : choix, [] : optionnelle, ||| : entrelacement, || : parallèle, * : répétition

Figure 3.15 – Représentation hiérarchique du scénario : «errance nocturne»

Tableau 3.5 – Résultats de la vérification des assertions pour le scénario 4

Portée	Assertion acyclique		
	Variables	Clauses	Temps (ms)
3	32232	107 448	220
4	43013	143363	372
5	53806	179298	157
10	146148	423971	707
15	498400	1353240	1482

3.9. CONCLUSION DU CHAPITRE

est incapable de mettre en évidence tous les cas de tests possibles. Cette lacune est fondamentalement liée à son paradigme. Nous proposons d'utiliser les modèles formels pour bâtir les scénarios nécessaires au bon fonctionnement des systèmes d'assistance ambiants.

Le modèle formel permet de capturer les propriétés d'un domaine pour que celui-ci reste cohérent et correct. Ainsi, il apporte une sémantique c'est-à-dire qu'il identifie les ambiguïtés qu'elles soient bonnes ou pas. Le modèle formel exprime les faits sous forme de règles. Les outils utilisés permettent de vérifier la conformité des spécifications en cherchant des contre-exemples dans une portée donnée, plutôt que de tester tous les cas possibles.

Ce chapitre présente la démarche menée pour spécifier de manière formelle les scénarios d'assistance. Il commence par définir les concepts qui oscillent autour des scénarios. Puis des exigences de la multidisciplinarité du sujet traité, il fait ressortir les caractéristiques et les attributs des scénarios. Il propose par la suite une organisation hiérarchique de représentation et de modélisation des scénarios pour l'assistance ambiante. Il montre par des exemples comment construire un scénario en mettant l'accent sur les questions à se poser pour chaque méthode (ascendante, descendante, combinaison). Quatre scénarios sont décrits et sont utilisés pour tester le modèle et la spécification principale proposés. L'outil ALLOY est utilisé pour spécifier, vérifier et assure la cohérence des quatre scénarios proposés.

Certaines limites ont été identifiées durant la production des spécifications avec ALLOY. La première vient du fait que la sémantique que nous proposons peut ne pas être la meilleure vu que nous utilisons une structure hiérarchique. La seconde vient du fait que le code produit par la spécification est verbeux et difficile à lire pour toutes les parties prenantes. Pour prendre en considération le deuxième inconvénient, nous avons proposé une approche de conception itérative et incrémentale où les spécifications générées automatiquement sont directement vérifiées. La suite du document montre comment le modèle hiérarchique de scénarios est couplé avec d'autres entités intervenantes dans le processus d'assistance pour offrir plus de services sensibles au contexte de l'utilisateur. Les ontologies sont utilisées pour faciliter cette capacité.

Chapitre 4

Modèle Sémantique des Systèmes Ambiants

Pour concevoir et mettre en place un système d’assistance ambiante DIY dédié à des personnes ayant peu ou pas de connaissances en informatique, nous proposons une ontologie de domaine et d’application. Cette ontologie structure et organise les connaissances du domaine de l’informatique ubiquitaire, de la sensibilité au contexte des capteurs et de l’environnement de la personne. Ceci permet aux agents logiciels de haut niveau de raisonner sur des faits plutôt que sur les données à l’aide des moteurs d’inférences ontologiques. Cette section présente la démarche, les contraintes, les conceptualisations et les résultats qui ont conduit à la définition d’un modèle sémantique d’assistance ambiante. La contribution proposée a directement été intégrée au projet principal NEARS.

4.1 De la donnée à la connaissance sémantique multiniveaux

Les habitats deviennent intelligents quand ils sont capables de comprendre et traiter les informations issues des capteurs, de l’environnement et de l’écosystème qui entoure l’habitat. Il faut donc mettre en place un processus rigoureux capable d’extraire l’information dans cet écosystème afin de le rendre disponible pour exploitation.

4.1. DE LA DONNÉE À LA CONNAISSANCE SÉMANTIQUE MULTINIVEAUX

Les systèmes d'assistance ambiants n'utilisant pas les ontologies sont, pour la plupart, basés sur des règles pour offrir l'intelligence nécessaire à l'habitat. Concrètement, ces systèmes utilisent la logique de prédicat de premier ordre pour trouver la contrainte à satisfaire. Chaque contrainte sous forme de programme (script) planifié écoute les entrées-sorties, récupère la valeur ou la donnée postée puis la propage à l'ensemble de ses règles pour vérifier celles qui seront satisfaites. Pour bien comprendre cela, reprenons l'exemple décrit plus haut dans la section sur la structuration des informations dans l'ontologie 1.3.1. Dans cet exemple, l'habitat intelligent souhaite éclairer le couloir lorsque la personne âgée se lève pendant la nuit. Si le capteur de mouvement de la chambre à coucher indique "ON" il déclenche la règle permettant d'allumer les lumières de la chambre dès qu'un mouvement dans la chambre est détecté. Cette règle pour être efficace ajoute des informations temporelles afin de ne s'exécuter qu'après 18 h. Malgré cela, cette règle reste très inefficace, en ce sens qu'elle ignore tout du contexte dans lequel elle s'exécute. Aucune connaissance sur l'historique du capteur n'est connue du système de règles, ce qui engendre une incapacité à établir des règles très précises. De plus, il n'est donné aucune information sur les interactions entre les règles existantes. Entrent-elles en conflit ? S'annulent-elles ? Le manque d'informations sur les autres entités qui interviennent dans le processus ne permet pas de rendre cette règle robuste ni de gérer les informations temporelles et spatiales.

La Figure 4.1 compare les raisonnements établis par les systèmes qui utilisent ou non les ontologies pour raisonner sur la détection de mouvement dans la chambre à coucher ou sur toutes autres activités. Les ontologies sont basées sur la logique de description et l'Internet des objets se construit mieux dans un environnement (domaine) où la connaissance est ouverte. Cette construction facilite l'interopérabilité et la réutilisabilité des concepts, pour permettre un accès plus large à la connaissance ambiante. Les systèmes de raisonnement basés sur la programmation logique peuvent donc utiliser cette connaissance pour offrir une inférence à un niveau contextuel mieux enrichi par la connaissance ambiante. Dans l'exemple de la figure 4.1.a), le système n'utilise pas l'ontologie, du coup le raisonnement lié à la règle « mouvement dans la chambre » se restreint à cette information, sans inférer l'activité que réalise la personne par manque d'informations contextuelles. Par contre, lorsque le raisonnement

4.1. DE LA DONNÉE À LA CONNAISSANCE SÉMANTIQUE MULTINIVEAUX

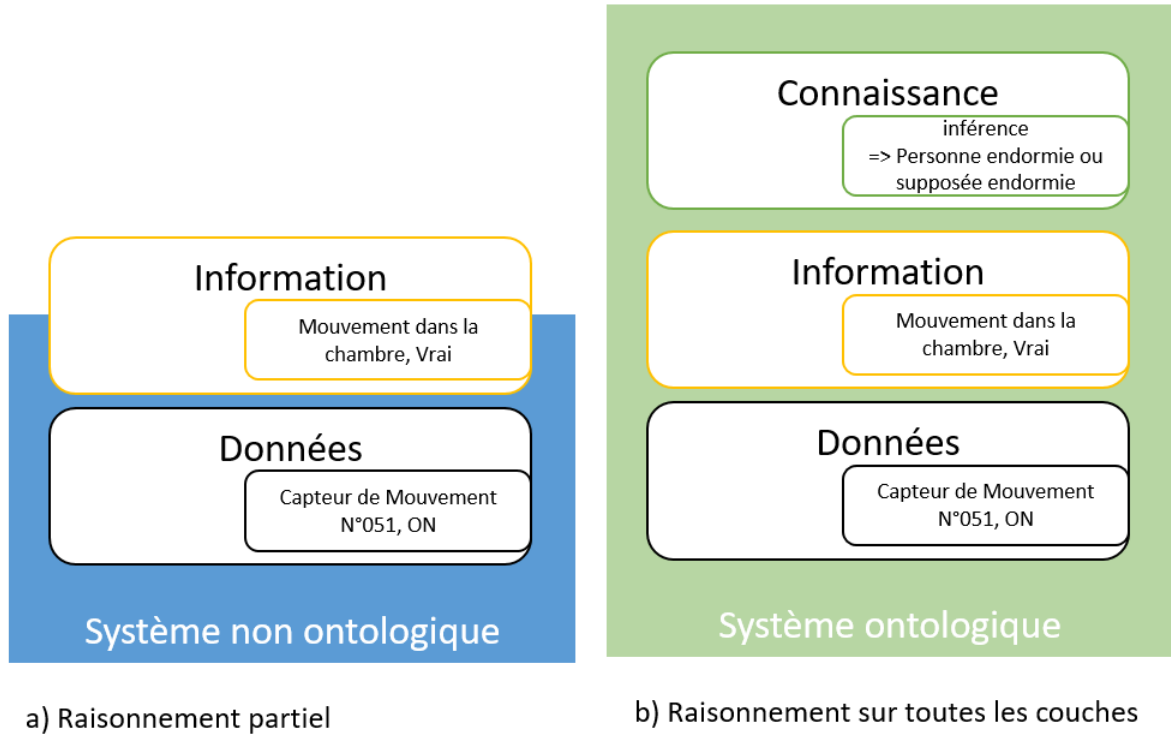


Figure 4.1 – Intégration de la connaissance dans le raisonnement

est basé sur les ontologies, les différentes couches de connaissances sont impliquées. Ainsi, le système infère non seulement que la personne est dans la chambre, mais aussi peut inférer son activité, à savoir qu'elle est endormie, en ajoutant d'autres informations au contexte : l'heure, les états d'autres capteurs. (voir Figure 4.1.b)

Cette figure permet de comprendre le positionnement et justifie en même temps le choix des ontologies pour enrichir les systèmes de raisonnement basés sur les règles. Cependant à notre connaissance, aucune solution ontologique existante ne propose de solution complète basée sur l'approche DIY. Le développement de l'ontologie des habitats intelligents proposée est composé des étapes suivantes : conceptualisation, spécification sémantique et formalisation. Nous avons opté pour un cycle de développement itératif et incrémental, principalement à cause de la multidisciplinarité de notre équipe.

4.2. LA CONCEPTUALISATION DE L'ONTOLOGIE

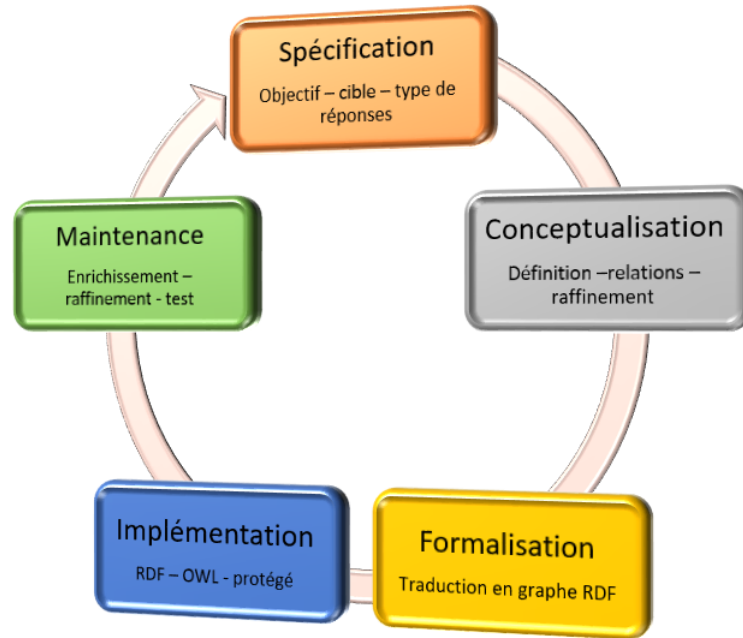


Figure 4.2 – Étape de développement de l'ontologie

4.2 La conceptualisation de l'ontologie

Le but de cette étape est d'abstraire le modèle de représentation et de structuration de la connaissance ambiante par rapport aux éléments et entités qui les constituent. Elle utilise les informations et la connaissance obtenue des travaux de Robert Radziszewski étudiant en maîtrise au laboratoire DOMUS [88, 89]. Le but des travaux antérieurs était de construire un prototype d'assistance à l'errance nocturne. Les tests et l'expérimentation menée à domicile ont permis de raffiner le prototype. Plusieurs ontologies [95, 114] proches de l'intelligence ambiante ont été testées pour savoir celles qui se rapprochaient le mieux de nos exigences.

En nous inspirant de la méthode de développement des ontologies intitulée « *Methodontology* » [71], nous avons organisé la conception de l'ontologie autour de cinq étapes comme le montre la Figure 4.2 :

1. **La spécification** consiste à la définition des objectifs, des cibles et des usages quotidiens de l'ontologie. Pour y parvenir, plusieurs questions ont été posées.

4.2. LA CONCEPTUALISATION DE L'ONTOLOGIE

Entre autres, quels sont les types de réponses fournis par l'ontologie, quels sont les aspects prioritaires à prendre en compte et quels sont les acteurs principaux l'ontologie.

Notre ontologie a pour but de modéliser et de structurer les besoins d'assistance ambiante. Elle permet à des applications, des humains ou à une communauté DIY de collaborer pour construire et suivre des scénarios d'assistance pour favoriser le maintien à domicile. L'utilisation de l'ontologie est justifiée par le fait qu'elle offre une sémantique bien formée et qu'elle est adaptée pour fournir des réponses ouvertes aux questions qui requièrent plus d'informations. Elle apporte toute la connaissance dont le contexte a besoin pour rendre l'assistance plus proche de la personne.

2. **La conceptualisation** consiste à organiser étape par étape la connaissance en définissant les termes, en les raffinant, et en les liant les uns aux autres pour créer des relations sémantiques. Elle passe aussi par la création des instances et la mise en place des règles adossées aux différents concepts. Plus spécifiquement, nous avons procédé par étapes. Réutilisation des ontologies existantes, alignement des concepts sur des concepts existants, réutilisation de termes existants sur d'autres ontologies sont les principales opérations effectuées dans la première phase. La seconde phase consistait en la mise en place des définitions et des termes du glossaire et en la définition des classes et de leur hiérarchie. La définition d'un concept est la spécification du sens voulu et souhaité associé à un terme de façon objective et complète. Puis ont suivi la définition des attributs et l'élaboration du dictionnaire des classes. Nous avons terminé l'étape de conceptualisation par la définition des restrictions des propriétés et des cardinalités des classes et des relations entre classes.
3. **La formalisation** consiste à la traduction des connaissances en langage OWL.
4. **L'implémentation** consiste à l'utilisation d'outils pour mettre en place la formalisation c'est-à-dire l'expression en RDF/OWL des concepts. Dans cette étape, nous avons utilisé le logiciel Protégé. Évaluation, vérification et validation des concepts sont effectuées à cette étape.

4.3. L'IMPLÉMENTATION DE L'ONTOLOGIE

5. **La phase de maintenance** permet d'enrichir, de bonifier, de raffiner, de corriger et d'améliorer l'ontologie de manière itérative.

Pour concevoir notre ontologie, plusieurs exigences ont guidé nos pas.

L'exigence 1 consiste en l'identification des fonctions et des opérations nécessaire à la mise en place de chaque entité. Il s'agit de décrire dans des relations ontologiques et sous forme de graphe RDF toutes les relations qui existent entre les entités intervenantes dans une relation. Partant du scénario d'assistance, pour chaque tâche qu'il décrit, extraire toutes les interactions et les représenter dans l'ontologie.

L'exigence 2 exprime la prise en compte des modèles existants. Existe-t-il des modèles de représentation générique capables d'être facilement adaptables au concept DIY ?

L'exigence 3 impose la mise en place d'une explicitation claire des contraintes métiers. L'ontologie doit être capable de représenter et d'expliciter toute la diversité de la multidisciplinarité et de la pluridisciplinarité.

L'exigence 4 consiste en la structuration et l'agencement de la représentation des scénarios. Toute la mécanique et l'organisation hiérarchique des scénarios doivent être encodées et présentées dans l'ontologie pour permettre de faire de l'inférence.

L'exigence 5 apporte une restriction forte sur la définition des abstractions de haut niveau entre différentes couches métiers. Il s'agit de mettre en place des interfaces pour faciliter le passage d'un domaine métier à un autre.

4.3 L'implémentation de l'ontologie

Le but de cette étape est de montrer comment le modèle proposé a été implémenté. Notre modèle pour couvrir la totalité des exigences est un agrégat de plusieurs ontologies. Six ontologies ont été conçues puis mises en ensemble pour former le modèle définitif. La première ontologie représente et spécifie un habitat, la deuxième

4.3. L'IMPLEMENTATION DE L'ONTOLOGIE

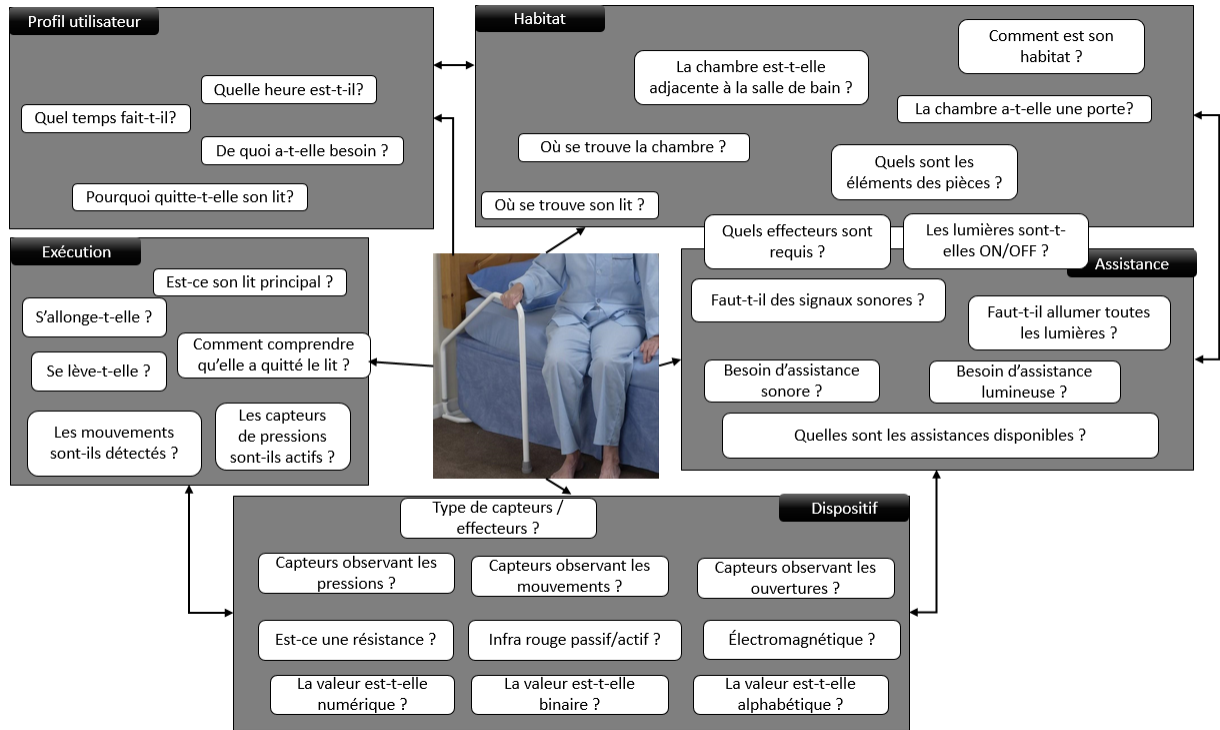


Figure 4.3 – Vocabulaires et questions autour d'une assistance ambiante pour le cas « sortir du lit »

les concepts du domaine des activités de la vie quotidienne. La troisième organise les scénarios, la quatrième traite des questions d'assistance à offrir. La cinquième structure et modélise les dispositifs (capteurs/effecteurs/contrôleur) nécessaires à la mise en place d'un environnement ubiquitaire. La sixième organise la connaissance contextuelle autour de la personne.

Partons d'un cas pratique pour mettre en évidence les relations existantes entre les ontologies construites. Considérons la situation : "la personne âgée sort de son lit pour se rendre à la salle de bain". La Figure 4.3 présente le vocabulaire et les questions que nous nous sommes posées pour modéliser l'assistance à cette personne. Ceci représente la base de notre processus de modélisation. Nous nous proposons de combiner une approche d'identification par thésaurus des termes et une approche à plat de structuration des connaissances pour modéliser l'assistance.

Le choix des terminologies est dirigé par les expérimentations effectuées, les besoins

4.3. L'IMPLÉMENTATION DE L'ONTOLOGIE

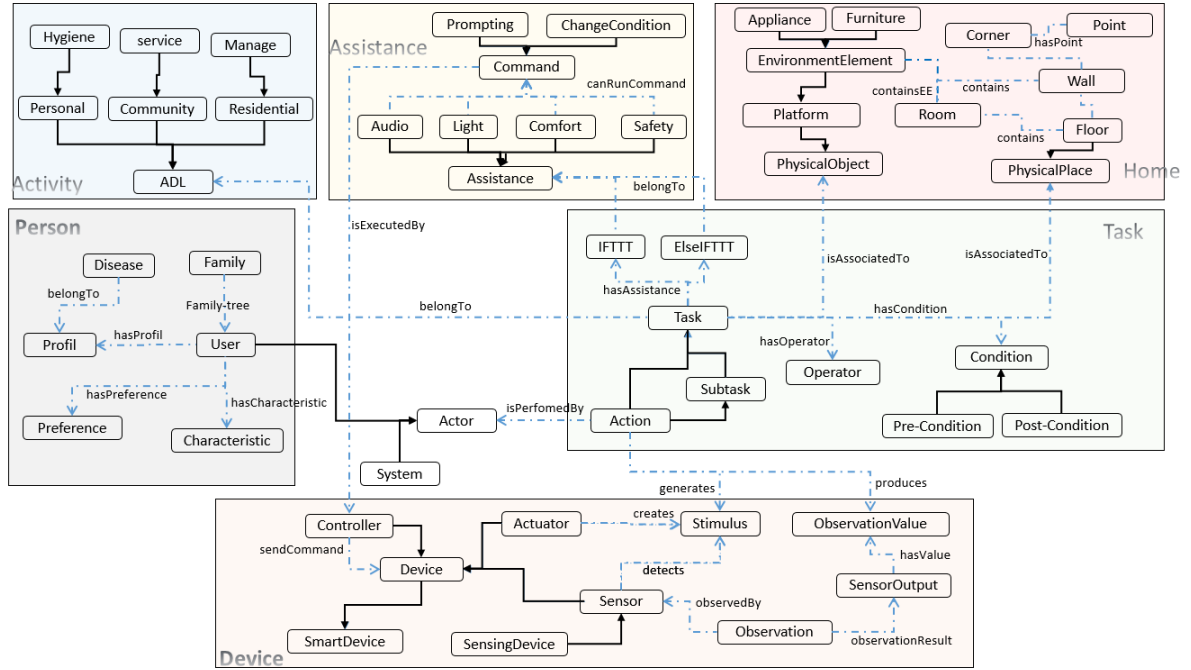


Figure 4.4 – Vue globale de l'ontologie

et les connaissances liées aux exigences métiers. Cette modélisation ontologique a pour volonté de donner la possibilité à l'utilisateur n'ayant pas de connaissances de décrire le comportement du système ambiant.

La Figure 4.4 présente une vue globale, mais incomplète, des interactions entre les différents concepts. Il est construit pour répondre aux questions inhérentes à la mise en place d'application sensible au contexte dans les habitats intelligents. Il s'agit entre autres des questions : qui ? Quoi ? Où ? Quand ? Comment ? Avec quoi ? Avec qui ?

Le modèle s'organise autour de six blocs. Chaque bloc est considéré comme étant une ontologie de domaine, car chacune capture toute la connaissance nécessaire à la modélisation d'un domaine métier particulier.

L'ontologie du domaine « *Home* » structure toute la connaissance nécessaire à la description d'un environnement (habitat) et des objets qui s'y trouvent. Elle permet entre autres de représenter les interactions entre les objets de l'habitat. L'ontologie du domaine « *Assistance* » représente des informations d'assistance que l'on peut

4.3. L'IMPLEMENTATION DE L'ONTOLOGIE

offrir à une personne dans un contexte d'intelligence ambiante. Elle structure les types d'assistance existants et offre des interfaces permettant de faire le pont avec les ontologies traitant des équipements. L'ontologie du domaine « *Activity* » définit la hiérarchie des AVQ. Comme nous l'avons très peu utilisée, nous n'en ferons pas de description exhaustive. L'ontologie du domaine « *Person* » est une ontologie qui traite des relations entre les personnes impliquées dans le processus. Nous avons importé une ontologie existante (family-tree ontology de Robert [106]) et nous avons ajouté les aspects cliniques touchant la personne. L'ontologie du domaine « *Task* » décrit et structure toute l'information sur l'organisation de l'activité de la personne dans son habitat. Elle définit et organise les concepts hiérarchiques de fonctionnement et de suivi d'une tâche. L'ontologie du domaine « *Device* » est une ontologie enrichie de l'ontologie SSN (« *Semantic Sensor Network* ») [27] à laquelle nous avons ajouté des capacités lui permettant de structurer en données mesurables et quantifiables l'activité d'une personne dans son environnement.

L'exemple de la Figure 4.3 se traduit par la modélisation disponible à la Figure 4.5. Les interactions pour le cas d'utilisation « sortir du lit » sont représentées par un parallélogramme, les autres formes représentent les classes. Ce diagramme met en lumière comment les liens se construisent pour faire ressortir les capteurs et effecteurs nécessaires à l'assistance.

La suite du document montre dans le détail comment les ontologies « *Task* », « *Device* », et « *Home* » sont structurées et interagissent entre elles.

4.3.1 Modélisation de l'environnement

La modélisation de l'environnement permet de définir le contexte des interactions dans l'habitat intelligent. Nous allons dans cette section définir la structuration sémantique de l'habitat afin d'en dégager les propriétés qui en résultent. Le but est de répondre aux questions d'ordre spatiales de la sensibilité au contexte. **Où suis-je ? Où vais-je ? Que puis-je utiliser ? De quoi ai-je besoin ?** La Figure 4.6 sous forme de graphe RDF montre les différentes interactions entre les classes de ce modèle. Cette représentation nous permet surtout de mettre l'accent sur les définitions clés de cette ontologie.

4.3. L'IMPLÉMENTATION DE L'ONTOLOGIE

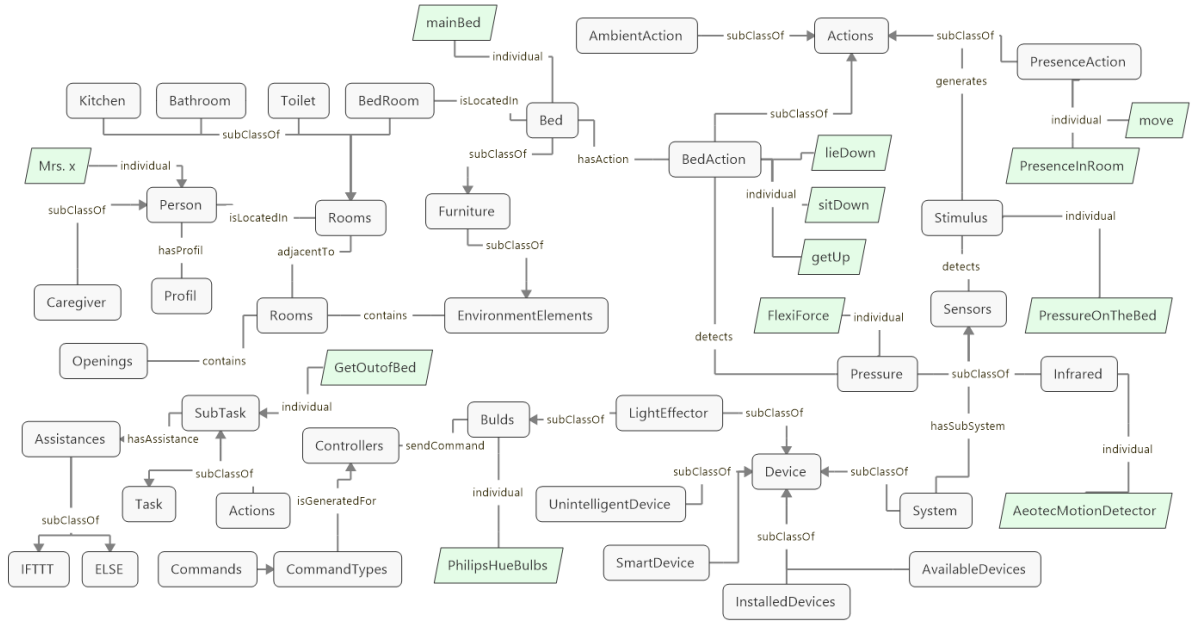


Figure 4.5 – Résultat de la modélisation du cas d'utilisation « sortir du lit »

La classe **Point** représente les coordonnées en trois dimensions d'une position dans l'espace dans un repère orthonormé x, y, z .

La classe **Corners** représente un coin ayant une position précise dans l'environnement. À ce titre, elle est liée à la classe point par la propriété **hasPoints**. Sémantiquement, la relation s'écrit sous la forme :

$$\text{Corners hasPoints only Points}$$

Cette relation traduit le fait qu'un corner ne peut se décomposer qu'en un seul point.

La classe **Wall** regroupe les différents coins d'un habitat. Elle est donc liée au concept de coins par la relation **hasListOfCorners**. Dans notre modèle, chaque mur ne peut avoir qu'exactly deux coins.

Cette définition est traduite par la relation :

4.3. L'IMPLÉMENTATION DE L'ONTOLOGIE

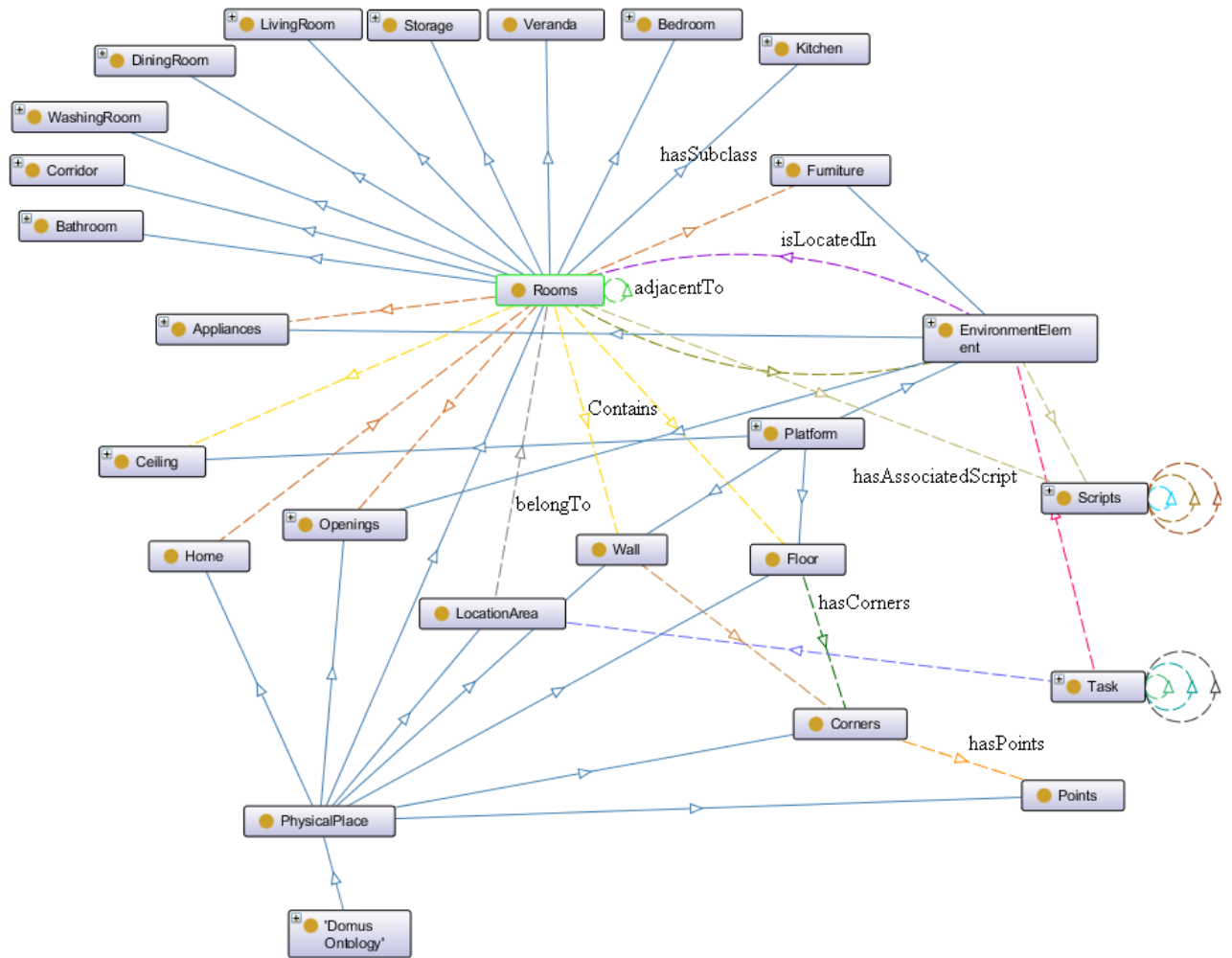


Figure 4.6 – Vue partielle de l'ontologie du domaine *Home*

4.3. L'IMPLEMENTATION DE L'ONTOLOGIE

```
(Contains some Appliances) and (Contains some Furniture) and (Contains some Openings)
and (adjacentTo some Rooms) and (containsEnvironmentElement some
EnvironmentElement) and (hasAssociatedScript some Scripts) and (Contains min 2 Wall) and
(Contains max 1 Ceiling) and (Contains max 1 Floor)|
```

Figure 4.7 – Spécifications formelles de la définition du concept de pièce

Wall hasListOfCorners exactly 2 Corners

La classe **Rooms** se définit comme un ensemble de deux murs minimum et éventuellement adjacent à une autre pièce. Elle contient un maximum un plafond, un sol et certains éléments de l'environnement (meubles, électroménagers) voir Figure 4.7. Une fois la définition faite nous construisons les relations existantes entre une pièce et une tâche. La relation **hasAvailableAction** permet pour sa part de lier les catégories d'actions que l'on peut effectuer dans une pièce.

Les autres classes et concepts qui sont rattachés à cette ontologie sont définis de la même manière. Ainsi, de fil en aiguille nous sommes capable grâce à cette organisation de prendre en compte les éléments en interaction pour contextualiser le lieu dans lequel se déroule un scénario ou une activité.

4.3.2 Modélisation des scénarios

Le scénario dans l'intelligence ambiante est la matière principale, c'est elle qui structure l'assistance à offrir. Tout au long de l'activation des ontologies pour suivre la personne âgée, c'est le scénario que l'on suit pour savoir ce que fait l'utilisateur, pour savoir comment il le fait, avec quoi il le fait et quand il le fait. Il permet de déterminer les outils qu'il faut mettre à la disposition de la personne pour satisfaire son but. Pour apporter des réponses aux questions liées à la sensibilité au contexte dans l'intelligence ambiante, nous avons structuré l'ontologie comme le montre la Figure 4.8. Les lignes pleines représentent les relations d'héritage qui existent entre les concepts. Celles en pointillés définissent l'association entre les concepts. Les couleurs permettent de discriminer les associations.

La contrainte majeure que nous avons durant cette phase était la traduction ou la transformation des AVQ en données pouvant être captées par des dispositifs

4.3. L'IMPLÉMENTATION DE L'ONTOLOGIE

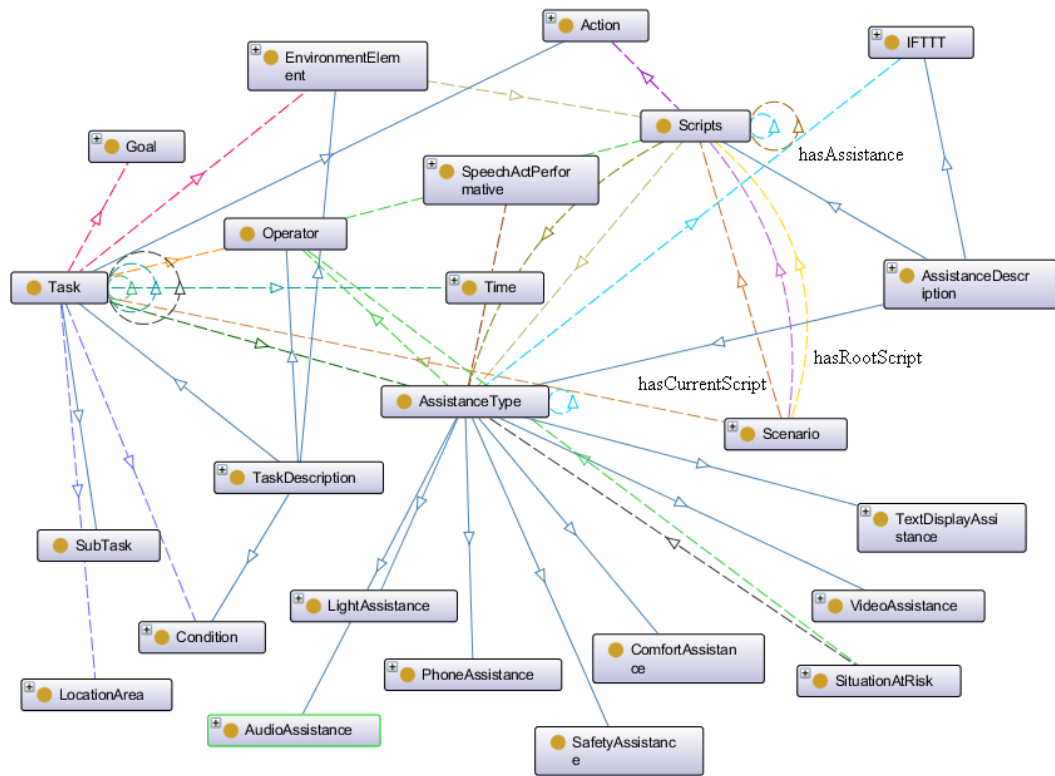


Figure 4.8 – Vue partielle de l'ontologie du domaine *Task*

4.3. L'IMPLEMENTATION DE L'ONTOLOGIE

```
(currentScript some Scripts) and (hasRootScript some Scripts) and (hasTask some Task) and  
(hasFrequencyUnit only Unit) and (hasTrigger only Scripts or Scenario)]
```

Figure 4.9 – Définition formelle d'un scénario

```
(currentScript some Scripts) and (hasAssistance some AssistanceType) and (hasChild some  
Scripts) and (hasElseAssistance some AssistanceType) and (hasListOfAction some Action)  
and (hasOperator some Operator) and (hasParent some Scripts) and (hasPredecessor some  
Scripts) and (isAssociatedToEnvironmentElement some EnvironmentElement) and  
(isAssociatedToRoom some Rooms)]
```

Figure 4.10 – Définitions formelles d'une tâche

électroniques d'informatique ubiquitaire. Pour comprendre la modélisation effectuée, il est nécessaire de passer par les définitions.

Un scénario est une collection de tâches. Il se caractérise entre autres par la tâche but, un déclencheur, l'instant de démarrage et sa fréquence d'exécution. Le déclencheur d'un scénario peut être un scénario ou une tâche. Dans cette conception, nous faisons la différence entre un minuteur qui annonce que le scénario doit commencer, puis une tâche qui déclenche effectivement le scénario. Pour illustrer cela, supposons que le scénario d'errance nocturne commence à 22 h (instant de démarrage). Cela signifie que le système à 22 h va surveiller le déclenchement de la tâche initiale qui démarrera le scénario d'errance nocturne. Pour le cas d'espèce, c'est la tâche "sortir du lit". La Figure 4.9 montre l'exemple de spécification produite pour le scénario.

Une tâche est une organisation hiérarchique de sous-tâches et d'actions. Pour simplifier l'implémentation, nous avons mis en place le concept de script qui encapsule à la fois la notion de tâche, assistance, tâche parente, opérateur et élément de l'environnement. Un exemple de spécifications de script est donné à la Figure 4.10.

Une action est un stimulus dans l'environnement pouvant produire une valeur observable. Donc une activité pour être comprise doit nécessairement déboucher ou commencer par une action. Chaque fois qu'une porte s'ouvre qu'une personne se déplace, qu'un objet est déplacé il y a un changement dans l'environnement. C'est ce

4.3. L'IMPLEMENTATION DE L'ONTOLOGIE

changement que nous appelons un stimulus et c'est lui qui est observable et parfois quantifiable.

Cette modélisation est en réalité une structuration dans l'ontologie de la spécification faite à la section sur la spécification des scénarios.

4.3.3 Modélisation des capteurs/effecteurs

De manière itérative et incrémentale, nous avons modifié l'ontologie SSN [27] et nous y avons rajouté des couches (classes, concepts, relations, entité, axiomes) pour prendre en compte les exigences de l'intelligence ambiante. L'ontologie SSN ne permet de définir que les capteurs et les événements de capteurs. Initialement utilisée pour des besoins météorologiques nous y avons ajouté toutes les spécifications pour la domotique et les systèmes ubiquitaires. Les Figures 4.12 et 4.11 sont des vues partielles de la nouvelle ontologie que nous proposons.

Les capteurs de cette ontologie surveillent les stimulus produits par les actions. Chaque stimulus converti en valeur observable est associé aux valeurs de sortie du capteur correspondant. Puis un concept abstrait est créé pour permettre le passage de la valeur observable à son processus de collecte d'informations. En conclusion, ce processus met en relation les capteurs impliqués dans une interaction.

La modification que nous avons apportée donne la possibilité aux effecteurs de créer des stimulus dans l'environnement, permettant ainsi aux capteurs de détecter les actions des effecteurs. On parle alors de boucle de rétroaction. Cette boucle se manifeste par les liens sémantiques existants entre les ontologies du domaine.

Les concepts principaux de cette ontologie sont les suivantes :

«**SensingDevice**» est la classe qui regroupe la totalité des types de dispositifs de collecte d'informations. On y retrouve entre autres les capteurs d'accéléromètres, de pressions, d'infrarouges, de contacts magnétiques...

«**EffectingDevice**» est la classe qui regroupe les types de dispositifs utilisés pour interagir ou donner des informations à la personne. Il s'agit entre autres des lumières, des écrans, des enceintes audio, etc.

4.3. L'IMPLÉMENTATION DE L'ONTOLOGIE

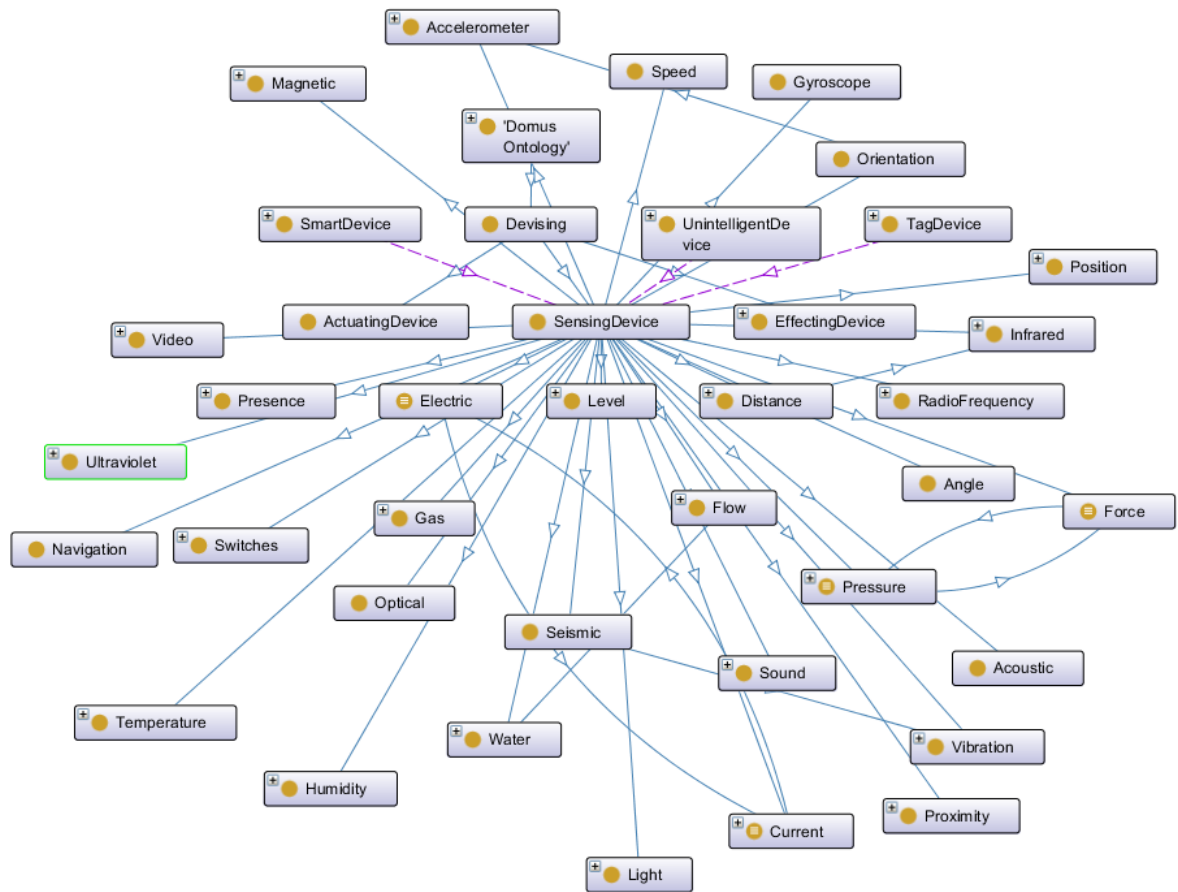


Figure 4.11 – Vue partielle de l'ontologie de domaine *Devices*

4.3. L'IMPLÉMENTATION DE L'ONTOLOGIE

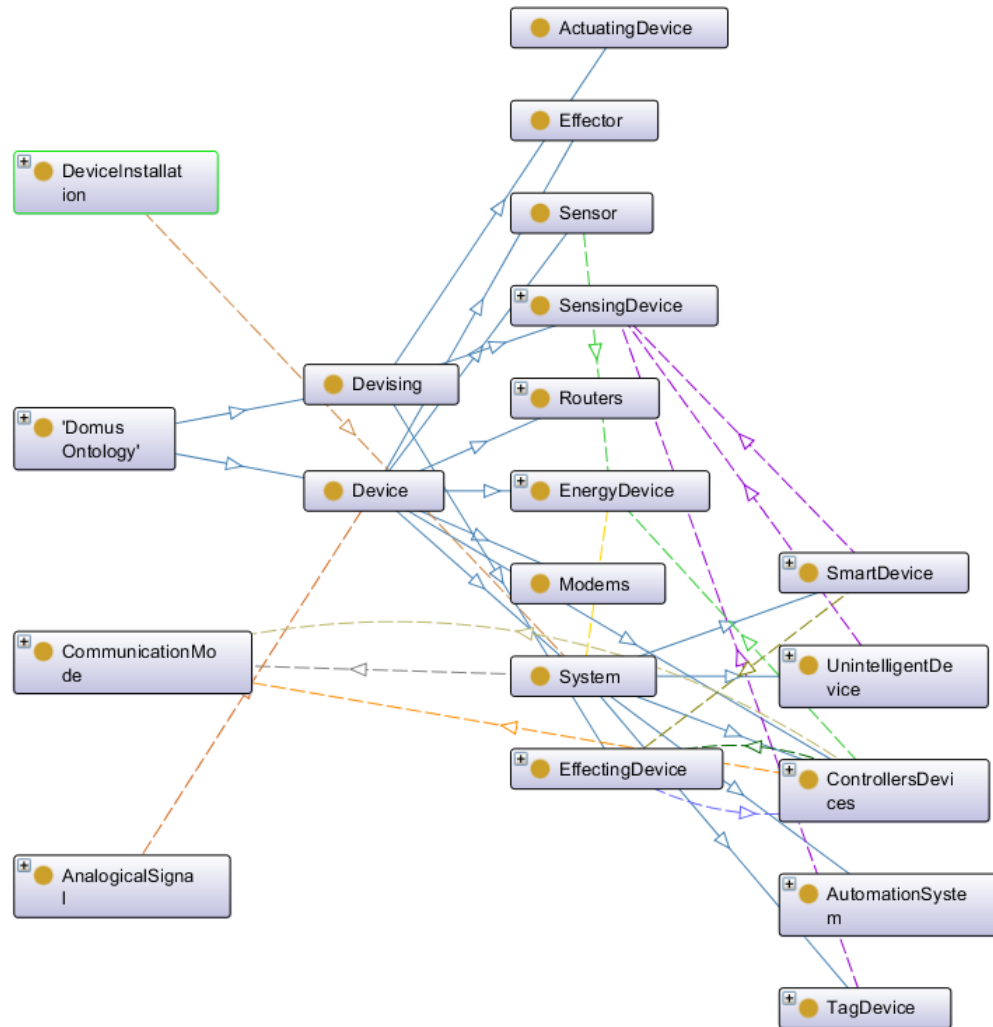


Figure 4.12 – Autre vue partielle de l'ontologie de domaine *Devices*

```
(canRunSoftware some Software) and (generatesCommand some Command) and
(hasCommunicationMode some NetworkCommunication) and (isGatewayForCommunication
some CommunicationMode) and (isPoweredBy some EnergyDevice) and
(manageCommunication some CommunicationMode) and (sendsCommandTo some Actuator)
and (sendsCommandTo some EffectingDevice) and (receivesCommandBy some Sensor)
```

Figure 4.13 – Définitions formelles d'un contrôleur

4.4. CONCLUSION PARTIELLE

«**SmartDevice**» est la classe qui regroupe les capteurs dits intelligents. L'intelligence ici est définie par rapport aux capacités de communication qu'offre le dispositif. C'est une classe disjointe de la classe «**UnSmartDevice**».

«**ControllerDevice**» est la classe responsable de la gestion et de la représentation des contrôleurs. Les contrôleurs sont généralement des unités de centralisation de la collecte d'information dans un réseau de capteurs. Le Figure 4.13 montre les formules logiques utilisées pour la définition du contrôleur. Cette figure illustre le fait que la définition d'un contrôleur est un dispositif pouvant exécuter un logiciel, puis qu'il génère des commandes, puis qu'il a un mode de communication, etc.

4.4 Conclusion partielle

Ce chapitre a présenté la mise en œuvre de l'ontologie conçue pour mettre en relation les informations sur l'habitat, sur le profil, sur la personne et ses contacts, sur les capteurs et sur l'assistance. Nous avons commencé par présenter la nécessité de raisonner à un niveau élevé, c'est-à-dire sur la connaissance plutôt que sur la donnée ou sur l'information. Ce procédé se fait en enrichissant les systèmes classiques de raisonnement par des liens sémantiques provenant d'une ontologie. Ce point est important pour comprendre le choix de l'utilisation de l'ontologie pour bâtir notre kit. Ensuite, nous avons abordé les aspects de conceptualisation de l'ontologie en insistant sur la façon dont nous avons spécifiée l'ontologie, sur l'utilisation des formalismes pour l'implémentation. Cette conception ontologique a tout le temps été guidée par cinq principales exigences . Le chapitre se termine sur l'illustration des différents modèles d'ontologies d'applications qui constituent l'ontologie du domaine présentée.

La conception de cette ontologie prend en compte les aspects importants de l'ingénierie de la connaissance à savoir : avoir une finalité applicative, être une ontologie interdisciplinaire, utiliser des approches symboliques pour la représentation, rendre les connaissances accessibles sous forme précise et en fonction du contexte et de l'opérationnalité, souplesse, modularité, compréhensibilité ...

Chapitre 5

Implémentation du Modèle Sémantique

Dans ce chapitre, nous allons présenter tour à tour les briques technologiques et l'API REST (REpresentational State Transfer) dédiées à la manipulation de l'ontologie. Nous nous intéressons principalement à l'implémentation, puis aux tests de l'application prototype développé dans le cadre du projet AGE-WELL pour offrir un ensemble d'outils et d'applications d'intelligence ambiante.

5.1 L'implémentation du modèle

Comme mentionné au chapitre précédent, notre modèle sémantique se base sur le couplage d'une ontologie de domaine et de spécifications formelles de conception de scénario.

Un ensemble d'outils composé d'un tableau de bord applicatif et d'un ensemble de dispositifs est en cours de développement au sein du laboratoire. Cet ensemble est basé principalement sur les contributions que nous avons apportées pour offrir la possibilité aux personnes ayant peu ou pas de connaissances en informatique de construire leur propre solution d'assistance ambiante. Le modèle que nous proposons permet de répondre à la question que se pose l'utilisateur : de quels capteurs ai-je besoin pour réaliser mon scénario ?

L'assistance construite par l'utilisateur dans le tableau de bord génère et propose

5.1. L'IMPLÉMENTATION DU MODÈLE

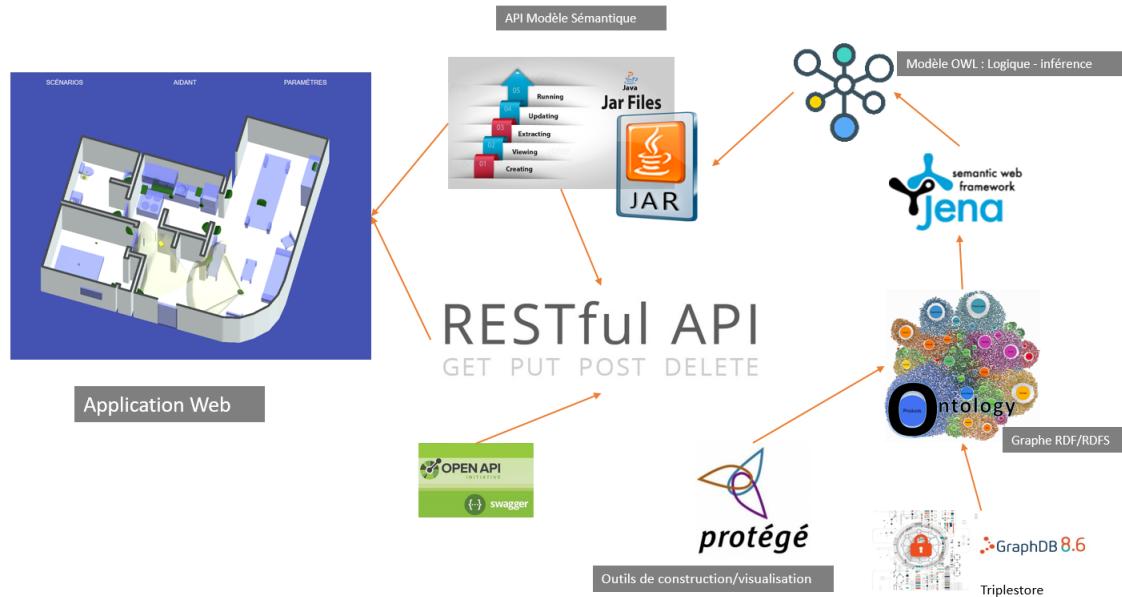


Figure 5.1 – Architecture fonctionnelle du modèle sémantique

une représentation visuelle de la connaissance ambiante et du contexte de la personne. La Figure 5.1 présente comment s'agencent les briques qui forment le modèle sémantique.

Le tableau de bord est construit autour d'un ensemble de services et de technologies du Web. L'idée est d'offrir un service disponible sur plusieurs terminaux et architectures hétérogènes. Le modèle sémantique est développé autour des technologies et outils suivants : Java, Owl, Jena, Protégé, HTML5 et nodejs. Le langage OWL est utilisé pour la spécification et la création de l'ontologie. L'API Jena est utilisée pour les requêtes et l'utilisation de l'ontologie via le langage de programmation Java. Une API REST est construite pour permettre l'utilisation de l'ontologie via le protocole HTTP. Toute la construction de l'ontologie s'est déroulée dans le logiciel Protégé.

Protégé est un logiciel libre et gratuit pour la création et la manipulation graphiques des ontologies. Il a été développé par le Stanford Medical Informatics (SMI) de l'Université de Stanford. Le logiciel est construit autour d'un ensemble de modules qui ajoutent des fonctionnalités supplémentaires pour la structuration et la construction de l'ontologie.

5.2. IMPLÉMENTATION DE L'API

L'ontologie à ce jour fait 9264 axiomes, avec 579 classes pour 190 propriétés (entité qualifiant une relation) de type objets et 60 de type données ont été construites. Un total de 1307 individus (instances) peuple l'ontologie.

5.2 Implémentation de l'API

Pour manipuler les données de l'ontologie depuis le tableau de bord une API REST a été développée. Cette API sert d'interface directe en HTTP ou HTTPS des fonctionnalités d'ajout, de modification, de suppression et de lecture des entités de l'ontologie. Comme pour tous les API REST les fonctionnalités sont accessibles en spécifiant la route et le verbe correspondant à l'action requise. Ces verbes d'action sont : *GET*, *POST*, *PUT* et *DELETE*.

5.2.1 Hétérogénéité des données et des applications

Pour faciliter la compatibilité entre les services et les applications, nous avons décidé d'utiliser le format JSON pour tous nos échanges de données. La Figure 5.2 représente une vue simplifiée du format de données utilisé pour la lecture via l'action GET de l'API d'un scénario. Le format présente la liste des pièces dans laquelle se déroule le scénario. La liste des tâches du scénario, la liste des assistances offerte, les éléments de l'environnement impliqués dans le scénario. On n'y retrouve aussi la fréquence d'activation du scénario et éventuellement la durée que met le scénario s'il est actif.

5.2.2 La conception par des procédés de méthodes agiles

Nous avons opté pour un travail sur la base des méthodes agiles de développement et de conception. Gestion SCRUM, outil Jira pour la création des demandes, GitLab pour le gestionnaire de version avec des mécaniques d'intégration et de déploiement continue. La stratégie principale de programmation était un peu de code, un peu de tests, un peu de refactorisation, un peu de tests, un peu de code, ainsi de suite. L'API est construite autour d'un ensemble de patrons de conceptions. Nous allons en citer quelques-uns.

5.2. IMPLÉMENTATION DE L'API

```
frequencyUnit : http://www.domus.usherbrooke.ca/diy-aide/ontologies/2017/2/automation#week
▶ rooms [0]
  description : {value}
  active : ☐ false
▶ rootTask {11}
  trigger : http://www.domus.usherbrooke.ca/diy-aide/ontologies/2017/2/automation#2049f4cd-ec
  type : {value}
  frequency : 1
  duration : 24
▶ assistances [0]
▶ environmentElements [0]
  name : Go to restroom
  id : http://www.domus.usherbrooke.ca/diy-aide/ontologies/2017/2/automation#2a76c6ac-b36d-
  starting : {value}
▶ actions [0]
▶ tasks [1]
  timeUnit : {value}
▶ currentTasks [0]
```

Figure 5.2 – Exemples de format de données JSON pour un scénario

Le patron de conception "état" a été principalement utilisé pour gérer, via les classes spécialisées, le comportement dynamique des instances des capteurs lorsque ceux-ci changent de valeurs.

Le patron de conception "stratégie" est utilisé pour les changements dynamiques du comportement de certains dispositifs qui embarquent plusieurs types de capteurs. Prenons par exemple le multi capteurs 6 en 1 de Fibaro, qui a des comportements de détection de mouvements, d'humidité, de vibration, de luminosité, d'ultraviolet et température.

Le patron de conception "décorateur" est utilisé pour ajouter des couches de contexte. Chaque fois que le contexte change parce qu'il y a un nouveau dispositif, un nouveau besoin ou une nouvelle tâche le contexte s'adapte automatiquement aux nouveaux éléments.

Le patron de conception "composite" est principalement mis en place pour la gestion hiérarchique des scénarios. En effet, la modélisation précédente des tâches comme une structure hiérarchique impose l'utilisation de la composition.

5.2. IMPLÉMENTATION DE L'API

```
PREFIX ssn: <http://purl.oclc.org/NET/ssnx/ssn#>
PREFIX family: <http://www.co-ode.org/roberts/family-tree.owl#>
SELECT (?action AS ?id) ?environmentElementType ?typeEnvironment ?name ?description ?type WHERE {
  BIND (<http://www.domus.usherbrooke.ca/diy-aide/ontologies/2017/2/automation#Bed> AS ?environmentElementType).
  ?environmentElementType rdf:type ?typeEnvironment.
  ?environmentElementType domus:hasAvailableAction ?type.
  ?type rdfs:subClassOf domus:Action.
  FILTER (?type != domus:AppliancesActions && ?type != domus:FurnitureActions && ?type != domus:SanitaryActions).
  ?action rdf:type ?type.
  OPTIONAL { ?action domus:name ?name. }.
  OPTIONAL { ?action domus:hasDescription ?description. }.
}ORDER BY ?name ?description ?type
```

Figure 5.3 – Exemple de requête SPARQL pour le choix des tâches

5.2.3 Implémentation sémantique

Nous nous appuyons sur la bibliothèque Jena pour la manipulation logicielle de l'ontologie. Le modèle sémantique obtenu en utilisant Jena est un modèle inféré. En effet, l'appel de la fonctionnalité « *built-in rule reasoner* » de Jena produit un modèle dans lequel toutes les relations sémantiques sont construites et les inférences faites. Ce modèle est plus performant qu'un graphe RDF et suivant le type de OWL choisi il offre une complétude maximale et décidable. Jena supporte plusieurs formats et il est simple de passer d'un format à un autre. Nous avons dans ce travail utilisé deux types de raisonneurs. Le transitif qui implémente toutes les relations transitives et les propriétés réflexives de RDF et RDFS. Il s'agit des relations impliquant les propriétés « *rdfs : SubPropertyOf* » et « *rdfs : subClassOf* ». Le dernier raisonneur que nous avons utilisé est le « *generic rules reasoner* » permettant d'implémenter le chaînage avant, le chaînage arrière et les stratégies hybrides de raisonnement sur les règles.

Après plusieurs modifications, tests et validations de performance, nous avons opté pour l'utilisation des requêtes en SPARQL plutôt que d'utiliser des appels de fonctions natives à Jena. En effet, une des versions de construction de l'habitat de la personne par interrogation de l'ontologie avec les fonctions natives de Jena mettait environ quatre à cinq secondes pour fournir le modèle d'habitat. Les mêmes opérations effectuées en utilisant une requête SPARQL donnent moins d'une seconde. SPARQL est supporté par Jena, et il est à la fois un protocole de communication et un langage de requête pour la manipulation des entités d'une ontologie. Il permet comme le SQL de modifier, créer, lire, mettre à jour et supprimer des informations dans une ontologie. La Figure 5.3 montre un exemple de requête SPARQL de l'API, pour la

5.3. CONCLUSION DU CHAPITRE

lecture des informations sur une tâche d'un scénario.

5.3 Conclusion du chapitre

La modélisation de la connaissance ambiante implique de disposer de structures permettant de traduire les contraintes d'assistance ambiante en une représentation formelle sans ambiguïté. Plusieurs outils produisent des spécifications cependant, seules les ontologies permettent d'évoluer dans un environnement ouvert. La suite du document présente comment l'ensemble d'outils utilise le modèle sémantique pour être sensible au contexte de la personne.

Chapitre 6

Architecture d'Exécution du Contexte décrit dans le Modèle Ontologique

L'intelligence ambiante est un domaine fortement multidisciplinaire, par conséquent il est assujéti à des interactions continues et à une intégration transversale. Pour tirer avantage des systèmes ambiants, les parties prenantes doivent s'assurer que les besoins des personnes soient pris en compte et que la technologie respecte chacune des exigences métiers. De manière logicielle, la réponse réside dans la mise en place d'une architecture logicielle capable de prendre en compte les évolutions dans le temps et dans l'espace. La sensibilité au contexte permet justement de prendre en considération ce type de problème.

Le fondement principal de l'intelligence ambiante est l'interprétation faite par le système des différentes composantes qui interagissent dans l'environnement et de la perception qu'elle a d'elle-même. Ce fondement s'appelle le contexte. L'objectif de cette section est de présenter notre modèle de contexte. Il s'agit de proposer une architecture logicielle pour la mise en place de contexte ubiquitaire pour l'assistance ambiante par des personnes ayant peu ou pas de connaissances en informatique. Notre objectif est d'utiliser le contexte comme support pour le développement, le déploiement et l'installation d'application DIY.

Les objectifs des bricoleurs ou des utilisateurs finaux ou des parties prenantes étant

6.1. POURQUOI ÊTRE SENSIBLE AU CONTEXTE

le plus souvent divergents, nous nous proposons de donner une architecture simple, intuitive, en couche et adaptative. L'idée est de produire une représentation architecturale qui prend en compte l'utilisateur, l'environnement physique de la personne, les objets qui interviennent dans le processus et l'assistance à offrir.

6.1 Pourquoi être sensible au contexte

Pour éviter les ambiguïtés dans les solutions d'assistance ambiante proposées, il est conseillé d'utiliser le contexte pour enrichir le système de plus d'informations. La question fondamentale devient donc quelle démarche employer pour mettre en place une architecture de contexte et comment identifier les éléments à prendre en compte dans le traitement de ce contexte ?

Il n'est pas aisé de répondre à ces questions puisque l'énumération de toutes les solutions possibles pour un problème donné est indécidable, c'est-à-dire qu'il n'existe pas d'algorithmes qui prennent en entrée un problème et nous donnent la liste complète des solutions possibles en temps polynomial. Nous nous proposons dans ce cas d'offrir un cadre architectural permettant d'exprimer le plus globalement possible les situations qui font appel au contexte. Notre modèle doit donc être guidé par l'agencement des composants métiers intervenant dans le processus. Nous avons commencé par mettre en évidence l'ensemble des composants qui interviennent dans le système. Pour chaque composant, nous avons identifié ses interactions avec les autres et ses entrées/sorties.

6.2 Exemple d'application

Pour illustrer notre processus de création d'architecture sensible au contexte, considérons l'exemple d'un système développé au sein du laboratoire DOMUS de l'Université de Sherbrooke dans le cadre du projet AGE-WELL. L'objectif est d'accompagner durant la nuit une personne âgée à satisfaire un besoin. Cette personne est atteinte de la maladie d'Alzheimer et a des épisodes d'errances nocturnes. Le besoin à satisfaire peut-être de plusieurs types tels qu' aller au toilette, boire un verre d'eau, se relaxer, etc.

6.3. MODÈLE ARCHITECTURAL DE SENSIBILITÉ AU CONTEXTE

M. Georges a 89 ans et est un passionné de jeux de cartes. Il vit avec son épouse qui a 65 ans, qui est brigadière et qui s'occupe d'une sucrerie. Georges est atteint de la maladie d'Alzheimer, quand il se réveille dans la nuit pour un besoin il allume toutes les lumières va satisfaire son besoin et oublie de temps à autre de retourner se coucher. Il se fait du café, joue au solitaire et se met à déambuler dans la maison sans but précis. Dans la plupart des cas, son épouse est réveillée et est obligée de le convaincre pendant des heures de retourner se coucher. Cependant, durant la journée Georges va prendre plusieurs siestes, mais son épouse qui tient la sucrerie est constamment fatiguée, parce qu'elle est plusieurs fois réveillée par Georges.

Notre but est de modéliser cette situation afin d'identifier les actions de Georges et de mettre en place la mécanique pour que son environnement lui offre l'assistance nécessaire au moment nécessaire afin de rassurer et faciliter la tâche à son proche aidant. Il ressort de ce cas que l'assistance à Georges doit démarrer une fois qu'il sort de son lit et se poursuit tant qu'il n'est pas retourné se coucher. Quels sont donc les moments clés de cette situation que nous pouvons extraire pour créer notre modèle architectural ?

Pour répondre à la question du **où**, du **comment** et du **qui** et du **avec quoi** de la sensibilité au contexte, nous utilisons l'ontologie développée. Puis, pour répondre à question du **quoi** et du **pourquoi** nous utilisons la spécification formelle du scénario d'errance nocturne. L'ontologie fait l'énumération de l'ensemble des dispositifs électroniques, des meubles et des entités qui entrent en interaction avec le scénario d'errance nocturne. il modélise aussi la connaissance que nous avons des habitudes de Georges.

La Figure 6.1 donne un exemple de compositions des activités précédentes. La suite du document part de cet exemple pour montrer comment construire le cadre architecture d'exécution du contexte de Georges.

6.3 Modèle architectural de sensibilité au contexte

Ce que nous présentons dans cette section est un ensemble de modules de base génériques, qui peuvent s'emboîter pour offrir un cadre d'exécution du contexte. L'idée est d'offrir des couches architecturales réutilisables qui s'entrelacent, s'assemblent

6.3. MODÈLE ARCHITECTURAL DE SENSIBILITÉ AU CONTEXTE

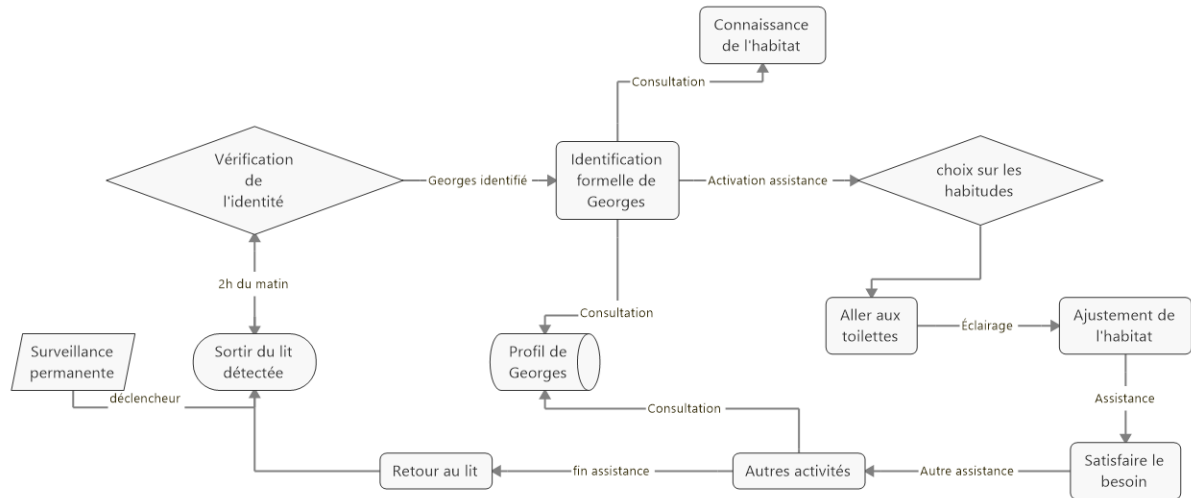


Figure 6.1 – Diagramme d'activité de Georges

pour permettre la construction d'application sensible au contexte. Notre proposition architecturale est indépendante de toute application.

L'architecture est composée de trois modules. Le premier module est celui de la collecte des informations issues des différentes sources (capteurs, personne, habitat, application.). Ce module est le plus bas dans la hiérarchie. Le second module est celui du contexte d'exécution et d'adaptation des informations collectées afin de les formater pour traitement et raisonnement ultérieurs. Il joue principalement le rôle d'intergiciel. Le dernier module est celui du gestionnaire de contexte. Ce module permet de faire l'interfaçage avec d'autres applications. La Figure 6.2 donne une représentation de cette architecture que nous avons construite et utilisée pour des expérimentations au laboratoire DOMUS.

6.3.1 Le Module de collecte de données

Ce module est considéré comme une sonde, car, c'est lui qui est responsable de collecter les données de diverses sources et de les rendre disponibles. Il est divisé en deux parties. La partie contexte de l'environnement et la partie Contexte de l'utilisateur comme le montre la Figure 6.2. Il est facilement extensible il suffit, pour cela

6.3. MODÈLE ARCHITECTURAL DE SENSIBILITÉ AU CONTEXTE

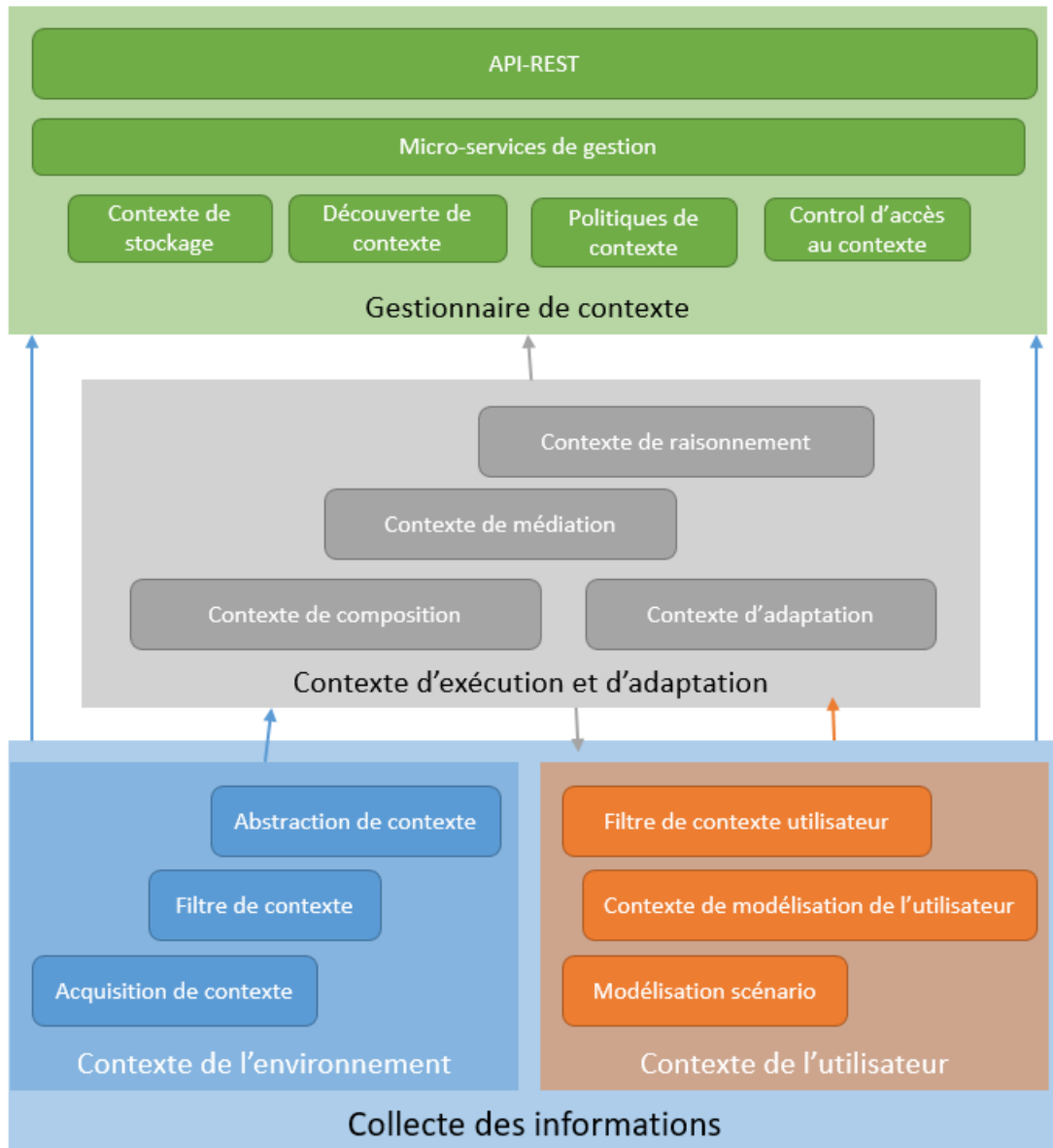


Figure 6.2 – Architecture logicielle pour les applications sensibles au contexte

6.3. MODÈLE ARCHITECTURAL DE SENSIBILITÉ AU CONTEXTE

d'ajouter une nouvelle source de données et de la connecter au module de niveau supérieur.

Le Contexte de l'environnement

Spécialisé dans la collecte des informations sur l'habitat intelligent, il surveille les capteurs et les dispositifs électroniques tels que les téléphones, les téléviseurs, bref toutes les sources de données disponibles et possibles.

Pour faciliter sa montée en charge, sa réutilisation et son entrelacement, nous l'avons organisé en trois couches. La couche d'acquisition de contexte, la couche de filtrage des données de contexte et la couche d'abstraction.

La couche d'acquisition de contexte

Elle collecte toutes les valeurs physiques de l'environnement. Elle est donc par ailleurs sensible aux différents stimulus créés dans l'environnement. Ces valeurs physiques sont soit une information collectée directement dans l'environnement via un capteur soit les résultats d'un traitement algorithmique.

Le processus de collecte des informations est un processus qui se fait en continu, nous conseillons donc d'avoir un contexte d'acquisition permanent. Grâce aux informations collectées, on doit pouvoir en tout temps être capable de localiser la personne, de savoir ce qu'elle fait, mais pas comment elle le fait ?

Ainsi dans notre exemple avec Georges, le contexte d'acquisition était constitué :

- Des capteurs de mouvement pour suivre en tout temps ses déplacements ;
- Des capteurs de pression pour savoir quand il quitte le lit ou y retourne ;
- Des capteurs d'eau pour la salle de bain ;
- Des capteurs électriques pour allumer certaines lumières et lampes de chevet ;
- Des capteurs de contact pour les ouvertures et les fermetures de porte ;
- Des capteurs de traversée de porte ;
- Un capteur porté sous forme de montre.

Tous les capteurs fonctionnent sous forme d'événements. Chaque fois qu'un changement survient, un événement est propagé à une unité de traitement que l'on appelle le contrôleur.

6.3. MODÈLE ARCHITECTURAL DE SENSIBILITÉ AU CONTEXTE



Figure 6.3 – Environnements d’acquisition de contexte dans un habitat intelligent

La Figure 6.3 montre un exemple de disposition de capteurs pour construire le contexte d’acquisition dans l’habitat du laboratoire DOMUS.

La couche de filtrage du contexte

Les données collectées par les sources hétérogènes sont envoyées via différents protocoles de communication aux contrôleurs. Ils sont chargés de la maintenance et des opérations de gestion de groupe dans le réseau. Puis, chaque donnée est étiquetée et acheminée à un intergiciel. Pour une meilleure gestion de l’hétérogénéité des données, l’intergiciel agrège toutes les informations, les étiquettes, les dates et les met à la disposition d’une autre couche pour des traitements ultérieurs.

Dans le cadre de l’assistance à Georges, nous avons mis en place les contrôleurs suivants :

- a)** Un contrôleur Vera 3 pour la gestion du réseau Z-wave.
- b)** Un contrôleur Philips Hue pour la gestion du réseau Philips utilisant le protocole ZigBee.
- c)** Un contrôleur Arduino pour le transport et la lecture des informations des capteurs ne supportant pas les protocoles sans fils.
- d)** Un contrôleur Bluetooth pour la diffusion des sons via des haut-parleurs sans

6.3. MODÈLE ARCHITECTURAL DE SENSIBILITÉ AU CONTEXTE

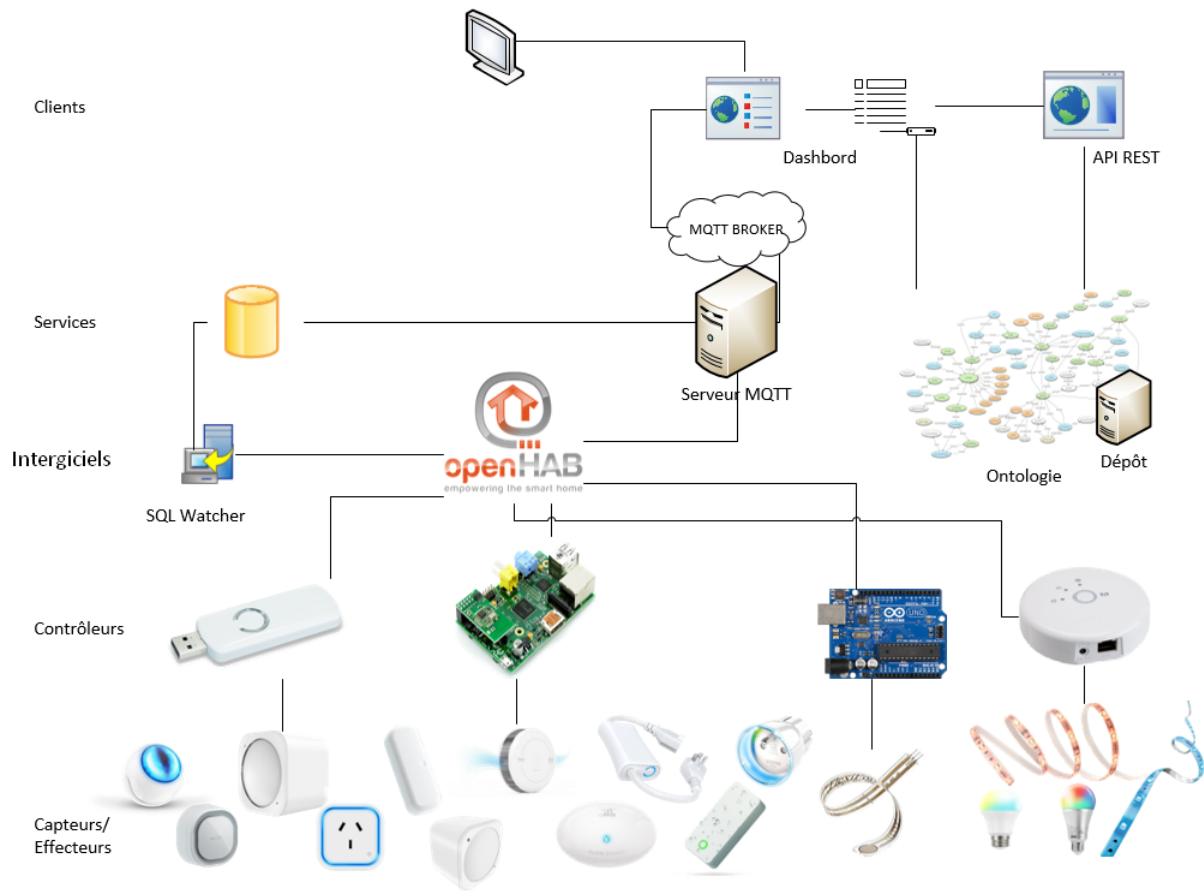


Figure 6.4 – Architecture physique du modèle de sensibilité au contexte

fil.

L'ensemble des contrôleurs étaient rattachés à OpenHab, un logiciel libre et gratuit permettant de gérer l'hétérogénéité des différentes sources de données. La Figure 6.4 montre une partie de l'interconnexion physique des contrôleurs et de l'intergiciel. Cet ensemble constitue donc le contexte de filtrage des données, en ce sens qu'il élimine toutes les informations de gestion et de synchronisation du réseau de capteurs.

La couche d'abstraction du contexte

La couche d'abstraction offre une visualisation simplifiée et compréhensible des informations issues de l'environnement. Elle fournit des outils permettant de diffuser ou de propager de manière spécifique les informations vers d'autres sources d'infor-

6.3. MODÈLE ARCHITECTURAL DE SENSIBILITÉ AU CONTEXTE

mation. Il s'agit spécifiquement de la mise en place d'un patron de type observer. En effet, plusieurs sujets peuvent vouloir être tenus informés instantanément des changements survenus sans pour autant être en mode demande active. Le rôle de ce module est de diffuser vers toutes les entités l'ayant sollicitée, les informations spécifiques de changement via les canaux réservés à cet effet.

Dans le cadre de notre exemple, nous avons utilisé le serveur et le protocole de communication MQTT (Message Queuing Telemetry Transport). Il utilise les notions de canal et de souscription pour tenir informés les sujets sur des changements. La Figure 6.4 montre comment se fait l'interconnexion entre MQTT et l'intergiciel.

Le Contexte de l'utilisateur

Il permet de renseigner l'application au complet sur le profil utilisateur, sur son état de santé, sur ses habitudes et sur ses préférences éventuelles. Cette source d'acquisition des données est justifiée par le fait que les personnes atteintes de la maladie d'Alzheimer vivent leur maladie de manière très différente, de sorte qu'une modélisation cas par cas est généralement recommandée. La connaissance du profil et des habitudes de vie de la personne est une information contextuelle importante voir capitale, car, elle permet de mieux répondre à la question comment et pourquoi. Une fois de plus pour faciliter la réutilisation, nous avons opté pour un découpage en couche. Ainsi, trois couches permettent d'exprimer les informations de ce module. La couche de modélisation des scénarios, la couche de modélisation du contexte utilisateur et la couche de filtrage de contexte utilisateur.

La couche de modélisation des scénarios

Cette couche décrit l'histoire de la personne ayant besoin d'assistance. Elle décrit de manière formelle le scénario ou comment la personne accomplit ses activités et où. L'approche de conception basée sur les scénarios est utilisée généralement dans ce cas et il a été longuement explicité dans les sections précédentes.

Dans notre exemple, Georges était à la fois co-créateur du scénario et co-concepteur de l'assistance, tout cela sous le contrôle des parties prenantes. Il nous a montré comment il réalise toutes les activités du scénario. Nous avons évalué avec les différents experts de chaque domaine comment, où et à quel moment nous jugions opportun d'assister Georges.

6.3. MODÈLE ARCHITECTURAL DE SENSIBILITÉ AU CONTEXTE

La couche de modélisation du contexte utilisateur

Ce processus itératif permet d'enrichir la base de connaissances avec de la connaissance sur l'évolution de la maladie de la personne. Elle prend aussi en compte le comportement de la personne face au système. Plusieurs informations sur le profil de la personne sont aussi prises en compte. Nous pouvons citer entre autres : les habitudes de vies de la personne, comment réagit la personne à un scénario ou une partie d'un scénario, à quel stade est sa maladie. . .

En ce qui concerne Georges, les tests cognitifs réalisés montrent qu'il est atteint d'Alzheimer à un stade modéré. Il ne fait jamais la cuisine et se couche tous les soirs à la même heure. Il pense qu'il se lève toutes les nuits ou presque pour aller aux toilettes. Il préfère souvent s'endormir sur son divan au salon. En plus nous savons qu'il aime jouer au solitaire et qu'il n'est pas à risque de chute.

La couche de filtrage de contexte utilisateur

L'objectif de cette couche est d'adapter la sortie des informations du module pour qu'elles correspondent aux capacités réelles de la personne. La couche de filtrage prend en entrée le profil, les habitudes, les contraintes et les préférences de la personne et retourne les contraintes nécessaires pour permettre à la personne de réaliser son besoin.

Georges a eu besoin d'indicateur lumineux et parfois d'instructions vocales pour l'accompagner durant ses épisodes d'errances nocturnes.

6.3.2 Le Module de traitement et d'adaptation de contexte

Responsable de l'exécution et du traitement des informations de contexte, le module de traitement de contexte est organisé en plusieurs couches. Le but des couches qui le composent est de fournir une interface aux informations du module d'acquisition. Quatre couches constituent ce module : le contexte de composition, le contexte d'adaptation, le contexte de médiation et le contexte de raisonnement.

Le contexte de composition

La composition de contexte permet d'avoir une vue standardisée des données des couches de base. Cette couche définit et opérationnalise les activités de fusion, de

6.3. MODÈLE ARCHITECTURAL DE SENSIBILITÉ AU CONTEXTE

compositions et d'agréations des informations pour produire ou construire des patrons de description de contexte. Les patrons obtenus peuvent être composés ensemble pour créer de nouveaux patrons.

La composition est le processus d'identification du lien fort qui existe entre des séquences d'événements. Elle crée une nouvelle séquence contenant les contextes liés.

Georges se réveille, fait des mouvements (s'incline pour s'asseoir sur le lit), sort du lit, passe sur le tapis tactile et marche vers la porte. Cette séquence se fait toujours dans cet ordre. La composition des différents éléments de cette séquence nous permet de créer un nouveau patron de contexte celui du réveil nocturne de Georges.

$$\begin{aligned} \text{Reveil Nocturne} = & (\text{heure} \geq 22h) \otimes (\text{Mouvement Lit} - ON) \otimes (\text{Tapis Sol} - ON) \\ & \otimes (\text{Pression Lit} - OFF) \otimes (\text{Mouvement Chambre} - ON) \end{aligned}$$

L'opérateur \otimes est un opérateur binaire associatif et non commutatif.

De la même manière, nous combinons différentes données pour créer des patrons de présences. Ces patrons de présences sont ensuite combinés entre eux pour construire de nouveaux contextes, comme l'illustre ce patron de contexte "sortie nocturne" :

$$\begin{aligned} \text{Sortie Nocturne} = & (\text{heure} \geq 22h) \otimes (\text{Présence Chambre} - OFF) \\ & \otimes (\text{Présence Salon} - OFF) \otimes (\text{Présence Toilette} - OFF) \otimes (\text{Présence Cuisine} - \\ & OFF) \otimes (\text{Présence Extérieure} - ON) \otimes (\text{Mouvement Escalier} - ON) \end{aligned}$$

La fusion est une opération qui rassemble plusieurs informations pour prédire ou estimer un nouvel état. Elle a pour but de réduire l'incertitude sur les informations résultantes. Le plus important ici c'est que peu importe l'ordre dans lequel les événements surviennent, la fusion donnera toujours le même résultat.

Dans le cas de Georges nous pouvons avoir un nouveau contexte en fusionnant les informations "pression sur le lit", "mouvement dans la chambre", puis "absence de pressions sur le tapis tactile au sol" qui donnerait le patron "présence sur le lit".

$$\begin{aligned} \text{Présence au lit} = \\ (\text{Mouvement Lit} - ON / OFF) \oplus (\text{Tapis Sol} - OFF) \oplus (\text{Pression Lit} - ON) \end{aligned}$$

L'opérateur \oplus est un opérateur binaire associatif et commutatif.

6.3. MODÈLE ARCHITECTURAL DE SENSIBILITÉ AU CONTEXTE

L'agrégation est l'opération utilisée pour appliquer un ensemble de fonctions mathématiques aux données. Il est utilisé pour lisser les données, calculer la moyenne, le seuil, le maximum, le minimum, la médiane, etc.

Georges est censé passer environ 10 h par nuit sur son lit. Il sort entre trois ou quatre fois par nuit pour des durées approximatives de 30 minutes maximum. Quand il boit un verre d'eau, il doit le faire en moins de 5 minutes. Un patron de contexte est mis en place pour prendre en compte cette situation.

Le contexte d'adaptation

Le contexte d'adaptation permet de construire la boucle de rétroaction. Cette couche se charge d'informer, la personne, le système et l'habitat des modifications survenues dans le contexte actuel. Cette rétroaction permet au système d'être au courant de son propre état et de s'ajuster en conséquence.

Chez Georges, elle nous permet de ne pas allumer le plafonnier quand la lampe de chevet est allumée ou de préférer la télévision quand elle est allumée plutôt que d'allumer les haut-parleurs Bluetooth.

Le contexte de médiation

Chaque patron de contexte généré par la couche de composition de contexte est envoyé pour sélection et prétraitement à un médiateur. Le rôle du médiateur de contexte est de choisir le patron à exécuter en fonction d'un certain nombre de critères. Il préempte les patrons de contextes selon différentes contraintes établies par les exigences de l'assistance à offrir, selon le profil de la personne, etc. Il est capable d'interrompre l'exécution d'une tâche précise pour lancer une exécution importante ou sécuritaire.

Dans l'assistance à Georges le choix se faisait entre les patrons "aller dans la chambre à coucher", "s'endormir dans la chambre", "s'endormir au salon sur son divan", "réveil nocturne", "retourner se coucher", "aller aux toilettes", etc.

6.3. MODÈLE ARCHITECTURAL DE SENSIBILITÉ AU CONTEXTE

Le contexte de raisonnement

Le patron de contexte choisi (élu, préempté) est transmis pour traitement à une unité spécialisée chargée d'analyser le contexte et de proposer la meilleure assistance correspondante. Cette couche s'appelle la couche de raisonnement de contexte. Elle valide l'intégrité des patrons et la structure des informations contextuelles associées. C'est donc un moteur d'inférence de haut niveau pour le raisonnement logique des règles d'assistances. Plusieurs techniques sont utilisées à ce niveau : la sémantique utilisant les ontologies et les SWRL (Semantic Web Rule Language), la logique floue, des réseaux bayésiens, la logique de premier ordre ou l'écriture de règles booléennes simples.

Dans l'expérimentation avec Georges, nous avons utilisé le système *Drools*. Ce système utilise un raisonnement déductif basé sur la logique de premier ordre pour surveiller et activer les règles métiers liées à l'assistance correspondant à la satisfaction des besoins de Georges.

Il est important de vérifier la présence de Georges dans la maison pour éviter que des personnes qui passent par là ou qui rendent visite à Georges ne déclenchent l'exécution de la règle.

Prenons une autre situation vécue par Georges. Il est 22 h et 10 minutes, le système de surveillance de scénario d'errance nocturne s'est activé 10 minutes plus tôt, mais il n'est pas encore déclenché, car il attend que Georges se couche sur son lit. À 22 h 20 le système ne détecte aucun mouvement dans la chambre, aucune pression sur le lit, mais localise Georges dans le salon, assis sur sa chaise berçante favorite à écouter la télévision. Suivant le profil de Georges, il est dit qu'il s'endort toujours à 22 h et qu'il a le droit de dormir sur son fauteuil. À ce moment, le système déclenche le scénario d'errance nocturne avec comme déclencheur le fauteuil qu'il faut surveiller. Le système crée directement une ambiance de sommeil en abaissant la luminosité, en éteignant la télé.

6.3.3 Le Module de gestionnaire de contexte

Responsable de la gestion, de l'organisation, de la maintenance, de la disponibilité et de l'accessibilité des contextes, le gestionnaire de contexte est la couche la plus haute

6.4. CONCLUSION PARTIELLE DU CHAPITRE

et celle par laquelle on accède aux autres modules. Il se présente comme un ensemble de micro-services. Des micro-services de persistances, de stratégies et de gestion de groupe sont implémentés. L'idée dans l'utilisation des micro-services c'est que ce sont des applications indépendantes les unes des autres et à couplages très faibles conçus pour effectuer une tâche unique. L'ensemble est orchestré à l'aide d'une API REST (REpresentational State Transfer). L'utilisation des micro-services facilite la gestion du déploiement et la maintenance du fonctionnement distribué des applications. Des exemples de micro-services à réaliser dans le gestionnaire de contexte sont :

- Le micro-service pour le contexte de stockage des données ;
- Le micro-service pour le contexte de découverte de nouveaux capteurs/dispositifs ;
- Le micro-service pour le contexte de découverte de nouveaux services ;
- Le micro-service pour les politiques d'accès aux informations de contexte ;

Pour Georges, nous avons mis en place un micro-service pour le stockage des données dans une base de données PostgreSQL. Nous avons mis un micro-service pour la surveillance et la génération d'alertes sur le niveau de batterie des capteurs et la défaillance du système.

6.4 Conclusion partielle du chapitre

La sensibilité au contexte est la démarche principale qui permet de décrire et de structurer la connaissance globale du système d'assistance. Nous avons proposé dans ce chapitre la mise en place d'une architecture logicielle en couche pour faciliter une utilisation DIY. En effet, les utilisateurs ne sont pas obligés d'agir au niveau des couches logicielles qui constituent l'architecture, mais plutôt au niveau des modules. Cette encapsulation offre un accès non technique au fonctionnement de l'architecture. Les données provenant de l'ontologie sont directement injectées dans l'architecture. Puis, leurs exécutions sont rendues transparentes pour les utilisateurs au travers des patrons de contextes. Les bricoleurs peuvent créer différents patrons de contextes, auxquels ils associent un contexte d'acquisition et un gestionnaire de contexte. Les changements souhaités peuvent se faire directement sur les modules concernés et non

6.4. CONCLUSION PARTIELLE DU CHAPITRE

sur la totalité de l'architecture. Cette abstraction en couche et en module est voulue pour encapsuler tous types d'application d'assistance ambiante.

Chapitre 7

Modèle de Positionnement des Capteurs

Dans cette section, nous allons nous inspirer du principe 7 de la technologie calme (Annexe A) qui énonce que : « La bonne quantité de technologie correspond au minimum nécessaire pour résoudre le problème ». Ceci offre à l'utilisateur une optimisation du nombre de capteurs nécessaires pour mettre en place son système d'assistance. Le chapitre présente dans sa première section la problématique du positionnement de capteurs. Puis, la section suivante présente le modèle que nous proposons.

7.1 La problématique de positionnement des capteurs

L'installation et le déploiement des services par-dessus des infrastructures de capteurs permettent de rendre les habitats intelligents. Pour minimiser les erreurs de positionnement des capteurs et avoir un bon calibrage des signaux émis, les capteurs sont généralement installés par des professionnels. Cette installation engendre parfois des coûts importants suivant les besoins de l'utilisateur et les exigences locales (disposition de l'habitat, facilité d'accès, nombre d'objets impliqué dans les interactions). Dans la plupart des cas, le scénario de base évolue, de nouvelles exigences apparaissent et parfois les capteurs ont un dysfonctionnement (épuisement des batteries,

7.2. LE MODÈLE DE POSITIONNEMENT

arrêt, mauvais calibrage, mauvaise lecture...). Dans ce cas, faire revenir le professionnel pour qu'il fasse un nouvel ajustement ou un nouveau déploiement implique des coûts supplémentaires et s'oppose à l'approche DIY.

Nous nous proposons d'accompagner l'utilisateur pour qu'il fasse lui-même ces ajustements. Dans ce chapitre, nous nous concentrons sur l'emplacement des capteurs. Le chapitre sur les ontologies a répondu à la question du type de capteurs. Nous proposons ici la définition d'un modèle mathématique pour le positionnement des capteurs dans un habitat intelligent. Nous allons montrer dans la suite du document que le problème du positionnement des capteurs consiste à optimiser le nombre de capteurs nécessaire pour assurer une meilleure couverture de l'assistance ambiante. Plus généralement, on parle de programmation linéaire.

7.2 Le Modèle de positionnement

7.2.1 Problème de positionnement

Sans nuire à la généralité, et pour faciliter l'explication, nous allons simplifier la problématique du positionnement en ne traitant que le cas des capteurs de mouvement. Le raisonnement pourra être généralisé aux autres types de capteurs.

Un capteur de mouvement est un dispositif électronique équipé généralement d'un détecteur de mouvements du type infrarouge passif. Chaque capteur a une portée de détections qui dépend du fabricant et de l'endroit où il est positionné. Par exemple, un "Aeotec Mutsensor 6 in 1" doit être positionné à une hauteur de trois mètres pour offrir une couverture de détection angulaire de 120° sur une distance de cinq mètres. Cependant, le "Fibaro Multisensor 4 in 1" pour la même hauteur et la même couverture angulaire donne environ sept mètres. Ces valeurs sont le plus souvent influencées par la position (le plafond ou le mur) du capteur dans l'habitat, la disposition de la pièce, les conditions environnementales de la pièce et ses encombrements. Les capteurs doivent au minimum couvrir toutes les pièces dans lesquelles ils sont déployés afin de garantir une bonne collecte de données. Concrètement, le problème revient à chercher comment maximiser la couverture par pièce tout en minimisant le nombre de capteurs à installer par pièce dans les conditions que nous imposent le type de capteurs, l'habitat et

7.3. LA MODÉLISATION DU PROBLÈME

$$\max_{(x_1, \dots, x_n)} \left[F(x_1, \dots, x_n) = c_1 x_1 + \dots + c_n x_n = \sum_{j=1}^n c_j x_j \right].$$
$$\left\{ \begin{array}{l} \bullet \text{ contraintes inégalités : } \forall i \in I_1, \sum_{j=1}^n a_{ij} x_j = a_{i1} x_1 + \dots + a_{in} x_n \leq b_i \\ \bullet \text{ contraintes égalités : } \forall i \in I_2, \sum_{j=1}^n a_{ij} x_j = b_i \\ \bullet \text{ contraintes de signes : } \forall j \in J_1, x_j \geq 0 \\ \bullet \forall j \in J_2, x_j \text{ de signe quelconque.} \end{array} \right.$$

Figure 7.1 – Définition formelle et générale d'un programme linéaire

l'utilisateur.

Comme ce problème s'apparente à un problème d'optimisation combinatoire, nous utilisons les outils de la programmation linéaire pour le résoudre.

7.2.2 Définition d'un problème d'optimisation combinatoire

On appelle programmation linéaire ou problème d'optimisation combinatoire. Le problème qui consiste à optimiser (maximiser ou minimiser) une fonction linéaire de plusieurs variables qui sont reliées par des relations linéaires appelées contraintes. Une définition plus formelle d'un problème d'optimisation linéaire est donnée par la Figure 7.1.

7.3 La modélisation du problème

Pour mieux comprendre notre problème, nous l'avons simplifié en assimilant la zone de détection d'un capteur à un cône comme le montre la Figure 7.2. Nous entendons par chevauchement de capteurs la zone commune observée par au moins 2 capteurs comme illustrée à la Figure 7.2

Plus simplement, il nous revient donc de déterminer le nombre de cônes à mettre

7.3. LA MODÉLISATION DU PROBLÈME

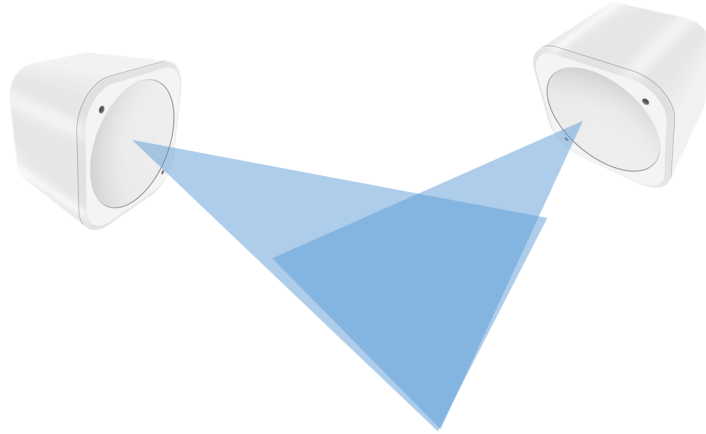


Figure 7.2 – Exemple de chevauchement de capteurs

dans une pièce pour la remplir. Cette opération se fera sous le contrôle d'un ensemble de contraintes de couverture de cônes et de leur chevauchement comme le montre la Figure 7.2.

La modélisation du problème de disposition des cônes entraîne les questions suivantes :

- Quelles sont les positions utilisables ?
- Quelles sont les contraintes du problème ?
- Que cherche-t-on à optimiser ?

Du problème de positionnement des capteurs, il ressort que nous pouvons créer deux types de modèles : un modèle de positionnement en 2D et un modèle en 3D.

7.3.1 Positionnement pour une détection en 2D

Dans le modèle de positionnement en 2D, nous faisons l'hypothèse que si la totalité du sol est couverte (exemple par des tapis tactiles) alors nous pourrions être satisfaits par la zone de couverture. Nos zones de détection se transforment en ellipse et notre problème se traduit par le tapissage du sol par des ellipses comme le montre la Figure 7.3.

7.3. LA MODÉLISATION DU PROBLÈME



Figure 7.3 – Positionnement des capteurs selon la version ellipse 2D

7.4. SPÉCIFICATION DU MODÈLE

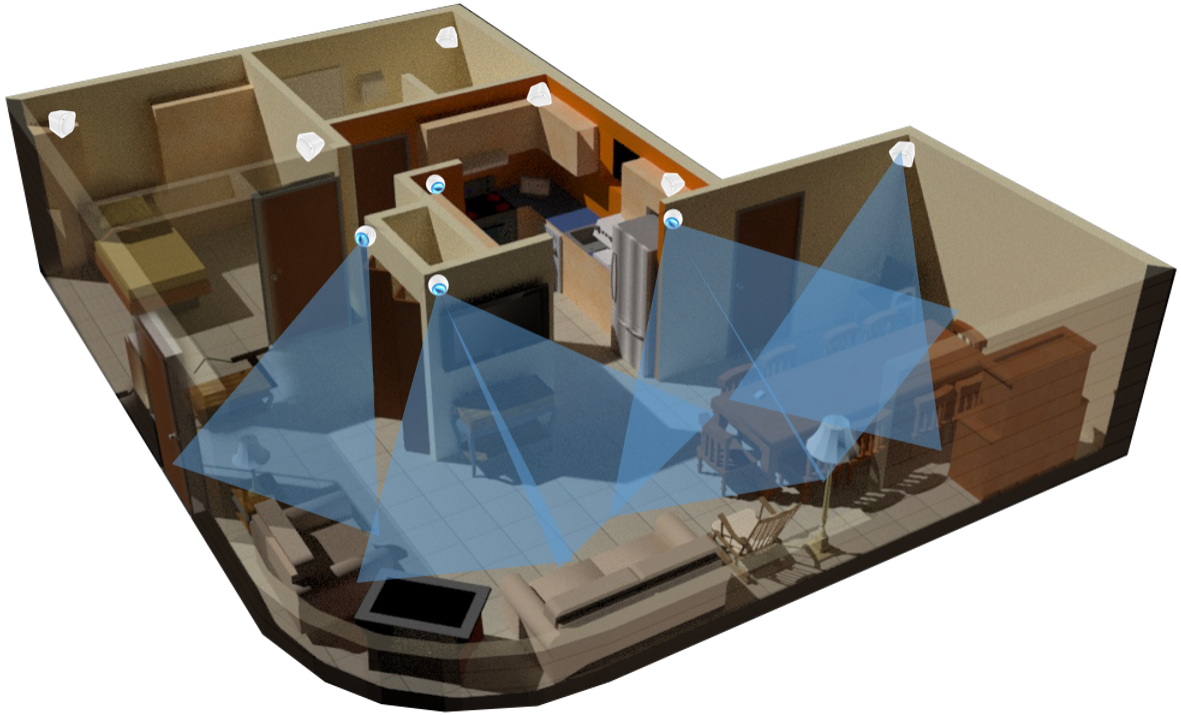


Figure 7.4 – Positionnement des capteurs selon la version en cônes 3D

7.3.2 Positionnement pour une détection en 3D

Le modèle de positionnement dans ce cas a pour objectif de détecter la totalité des mouvements qui surviennent dans la pièce, même ceux ne s'inscrivant pas dans un mouvement au sol, par exemple, un simple geste de la main. Plutôt que la surface couverte au sol nous nous intéressons au volume que les cônes couvrent dans la pièce (voir Figure 7.4). Dans cette configuration, les zones de chevauchements entre les cônes sont plus importantes et créent par la même occasion beaucoup d'interférence.

7.4 Spécification du modèle

Pour spécifier les contraintes de notre problème, nous allons écrire les équations linéaires qui lient chacune des contraintes à la prémisse de base. Cette prémisse concerne les positions utilisables et le nombre de capteurs qu'il faut pour couvrir la pièce. La construction des équations s'est faite au tour des questions suivantes :

7.4. SPÉCIFICATION DU MODÈLE

1. Quelles sont les positions utilisables ?
 - Il s'agit de donner en trois dimensions la position $x_i; y_i; z_i$ d'un capteur X_i . Ces éléments représentent les variables de notre problème.
2. Quelles sont les contraintes du problème ?
 - La zone de chevauchement entre deux capteurs est inférieure à un seuil connu ;
 - Le nombre de capteurs est limité ;
 - Le volume ou la surface de la pièce est connu ;
 - La position des capteurs est connue soit au plafond ou au mur.
3. Quelle est la variable à optimiser ?
 - Le nombre de capteurs sera minimal ;

D'où les équations linéaires suivantes :

Équations linéaires pour le positionnement en 3D

Le résultat de ces équations est de donner le nombre total de capteurs, tel que pour chaque capteur X_i on ait en trois dimensions la position $x_i; y_i; z_i$.

L'équation 7.1 traduit le fait que la couverture totale de l'ensemble des cônes doit être plus ou moins égale au volume de la pièce.

L'équation 7.2 permet d'exprimer le fait que le chevauchement entre les différents cônes doit être inférieur à un seuil donné.

L'équation 7.3 donne la valeur du volume de chaque cône.

Les équations 7.4, 7.5 et 7.6 donnent les distances euclidiennes des dimensions du cône.

X_i , traduit le vecteur qui est à la position $x_i; y_i; z_i$. Le vecteur x varie suivant la longueur de la pièce et y varie suivant la largeur de la pièce. Si le capteur est installé au plafond alors z est constant et toujours égale à la hauteur du plafond. Sinon il varie suivant la hauteur de l'installation sur le mur.

$$\min_{(x_i; y_i; z_i)} \left[\sum_i^n vol(X_i) = vol(\text{pièce}) + \epsilon \right] \quad (7.1)$$

7.4. SPÉCIFICATION DU MODÈLE

$$vol(X_i) \cap_{i \neq j} vol(X_j) \leq \eta \quad (7.2)$$

$$vol(X_i) = \frac{\pi R r h}{3} \quad \forall i \quad (7.3)$$

$$R = \sqrt{(x_{iR} - x_{jR})^2 + (y_{iR} - y_{jR})^2 + (z_{iR} - z_{jR})^2} \quad (7.4)$$

$$r = \sqrt{(x_{ir} - x_{jr})^2 + (y_{ir} - y_{jr})^2 + (z_{ir} - z_{jr})^2} \quad (7.5)$$

$$h = \sqrt{(x_{ih} - x_{jh})^2 + (y_{ih} - y_{jh})^2 + (z_{ih} - z_{jh})^2} \quad (7.6)$$

Équations linéaires pour le positionnement en 2D

Le résultat de ces équations est de donner le nombre total de capteurs, tel que pour chaque capteur X_i on ait en trois dimensions la position $x_i; y_i; z_i$.

L'équation 7.7 traduit le fait que la couverture totale de l'ensemble des aires doit être plus ou moins égale à l'aire de la pièce.

L'équation 7.8 permet d'exprimer le fait que le chevauchement entre les différentes aires doit être inférieur à un seuil donné.

L'équation 7.9 donne la valeur de l'aire de chaque ellipse.

Les équations 7.10 et 7.11 donnent les distances euclidiennes des dimensions de l'ellipse.

X_i traduit le vecteur qui est à la position $x_i; y_i; z_i$. Le vecteur x varie suivant la longueur de la pièce et y varie suivant la largeur de la pièce.

$$\min_{(x_i; y_i; z_i)} \left[\sum_i^n aire(X_i) = aire(\text{pièce}) + \epsilon \right] \quad (7.7)$$

$$aire(X_i) \cap_{i \neq j} aire(X_j) \leq \eta \quad (7.8)$$

$$aire(X_i) = \pi R r \quad \forall i \quad (7.9)$$

7.5. EXEMPLE DE POSITIONNEMENT DE CAPTEURS : LE CAS DU LABORATOIRE DOMUS

$$R = \sqrt{(x_{iR} - x_{jR})^2 + (y_{iR} - y_{jR})^2 + (z_{iR} - z_{jR})^2} \quad (7.10)$$

$$r = \sqrt{(x_{ir} - x_{jr})^2 + (y_{ir} - y_{jr})^2 + (z_{ir} - z_{jr})^2} \quad (7.11)$$

7.5 Exemple de positionnement de capteurs : le cas du Laboratoire Domus

Pour créer des instances de positionnement des capteurs, nous allons prendre le cas d'un capteur de mouvement : le capteur "*Aeotec by Aeon Labs MultiSensor 6*". La suite de cette section décrit des exemples de valeurs et de contraintes à soumettre au modèle de positionnement pour avoir des instances de positionnement de capteurs de ce type dans une pièce.

7.5.1 Description de la pièce

Considérons que nous voulions positionner des capteurs dans le salon, la salle à manger et le hall d'entrée du laboratoire DOMUS. Ces trois pièces sont communicantes comme le montre la zone en vert sur la Figure 7.5, l'ensemble formant la forme géométrique L.

Le calcul de l'aire de la partie verte donne $Aire(\text{pièce}) = 38.3m^2$. La hauteur du plafond est à $2.40m$, ce qui implique un volume de $91.92m^3$.

Suivant les caractéristiques du constructeur, les capteurs de mouvement respectent les propriétés suivantes. Pour une installation au plafond à une hauteur de $3m$ la sphère de détection a un petit diamètre de $6m$ et un grand diamètre de $8m$. La Figure 7.6, issue du manuel officiel, montre les dimensions disponibles pour une installation au plafond. La zone de détection possède un $volume(\text{capteur}) = 12\pi m^3$.

L'application des valeurs de l'appartement du laboratoire Domus au modèle de programmation linéaire pour une détection en 3D donné les résultats suivants :

$$\min_{(x_i; y_i; z_i)} \left[\sum_i^n vol(X_i) = 91.92 + \epsilon \right] \quad (7.12)$$

7.5. EXEMPLE DE POSITIONNEMENT DE CAPTEURS : LE CAS DU LABORATOIRE DOMUS

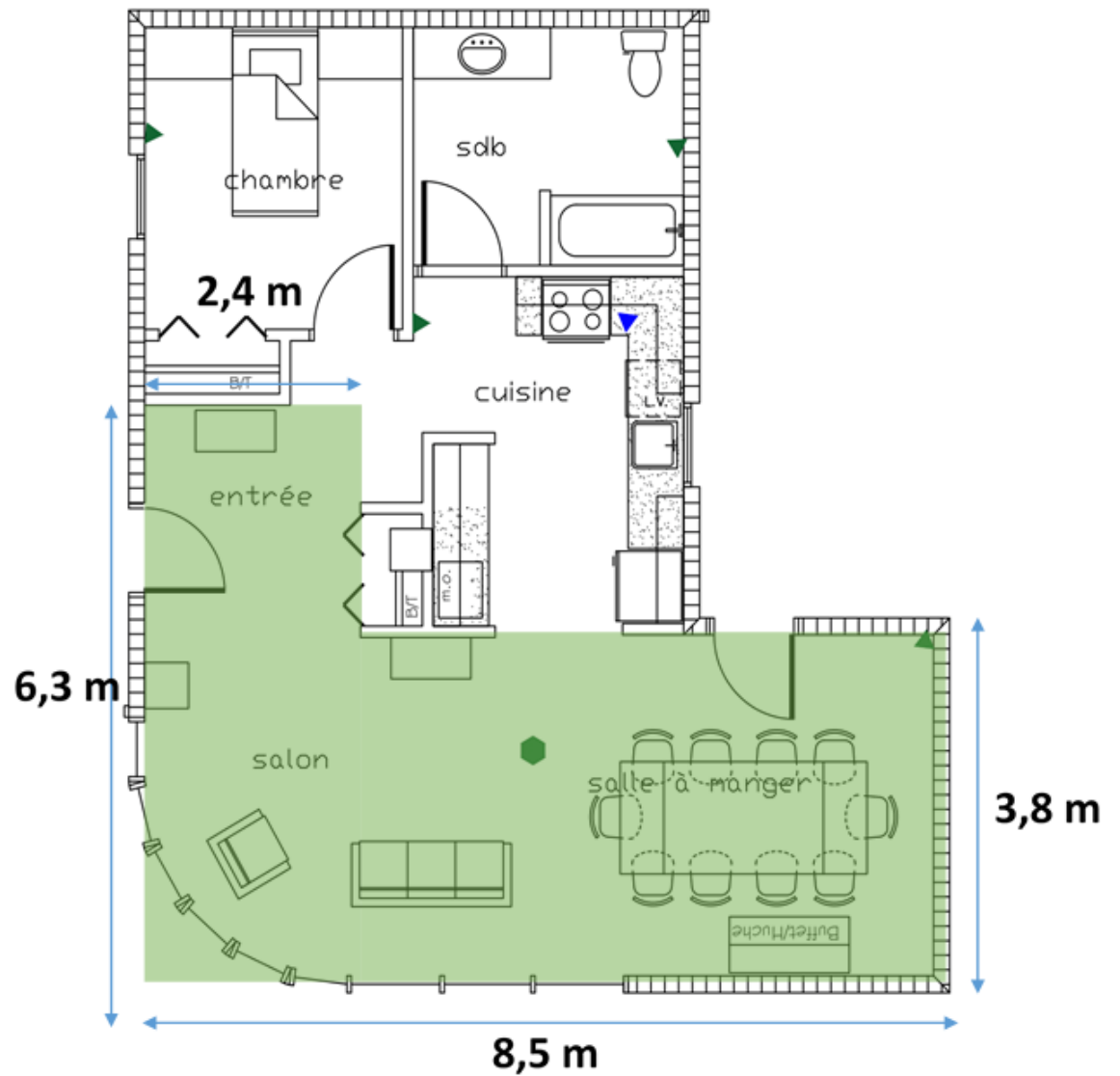


Figure 7.5 – Zone pour le positionnement de capteurs de mouvement

7.5. EXEMPLE DE POSITIONNEMENT DE CAPTEURS : LE CAS DU LABORATOIRE DOMUS

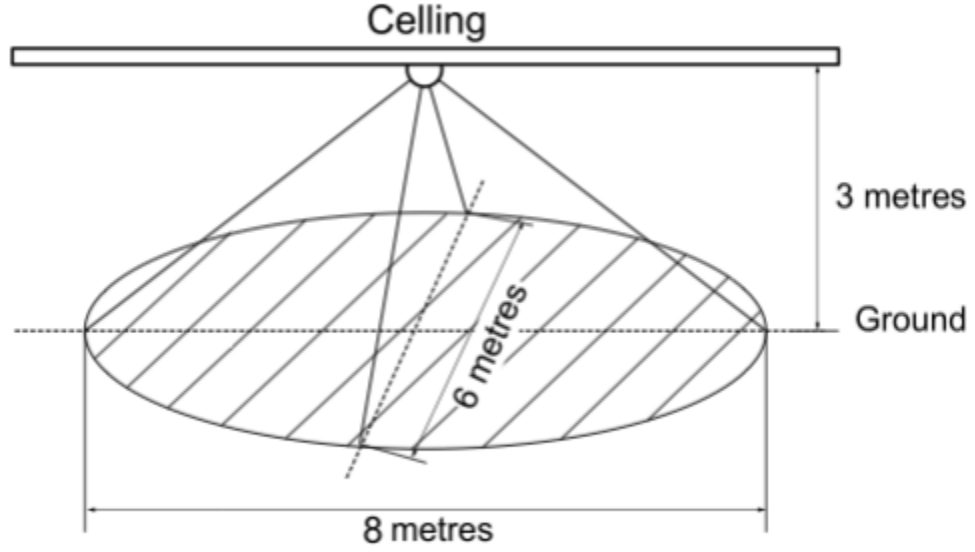


Figure 7.6 – Installation du capteur de mouvement au plafond : (source manuel officiel)

$$vol(X_i) \cap_{i \neq j} vol(X_j) \leq \eta \quad (7.13)$$

$$vol(X_i) = 12\pi \quad (7.14)$$

$$R = 4 \quad (7.15)$$

$$r = 3 \quad (7.16)$$

$$h = 3 \quad (7.17)$$

$$z = 3 \quad (7.18)$$

7.5. EXEMPLE DE POSITIONNEMENT DE CAPTEURS : LE CAS DU LABORATOIRE DOMUS

Si $\epsilon = 10$ et à $\eta = 2$ qui sont des valeurs expérimentales (utilisées pour se rapprocher de l'existant) alors le résultat de la programmation linéaire nous donne un nombre total de trois capteurs (2.7 pour être exacte) avec les positions suivantes $X_1(2.76, 3, 3.82)$, $X_2(0.34, 3, 6.33)$ et $X_3(5.94, 3, 2.23)$

Pour les mêmes valeurs l'application au modèle de programmation linéaire pour une détection en 2D donne les résultats suivants :

$$\min_{(x_i; y_i; z_i)} \left[\sum_i^n \text{aire}(X_i) = 38.3 + \epsilon \right] \quad (7.19)$$

$$\text{aire}(X_i) \bigcap_{i \neq j} \text{aire}(X_j) \leq \eta \quad (7.20)$$

$$\text{aire}(X_i) = 12\pi \quad (7.21)$$

$$R = 4 \quad (7.22)$$

$$r = 3 \quad (7.23)$$

$$h = 0 \quad (7.24)$$

$$z = 3 \quad (7.25)$$

Si $\epsilon = 10$ et à $\eta = 2$ qui sont des valeurs expérimentales (utilisées pour se rapprocher de l'existant) alors le résultat de la programmation linéaire nous donne un nombre total de deux capteurs (1.3 pour être exacte) avec les positions suivantes $X_1(1.0, 0.41, 2.06)$ et $X_2(3.09, 3, 7.52)$

De ces résultats nous avons l'impression que les instances en 2D, c'est-à-dire celles pour lesquelles seules les aires sont prises en compte semblent minimiser au mieux le nombre de capteurs (deux pour le cas d'espèce plutôt que trois en 3D). Il faut faire attention et prendre en compte le fait que placé sur le mur ils sont assujettis à des zones d'ombre entre $0.5m$ à $0.9m$ comme le montre la Figure 7.7. Pour l'assistance

7.6. CONCLUSION DU CHAPITRE

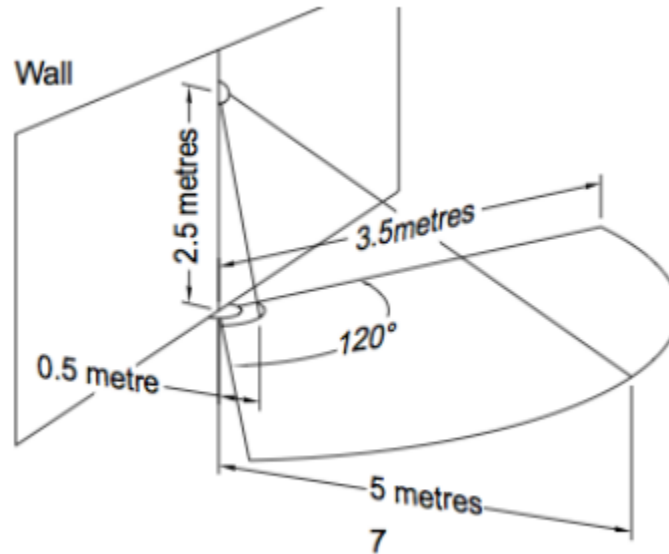


Figure 7.7 – Installation du capteur de mouvement au mur : (source manuel officiel)

ambiante il est important que l'orientation et l'angle d'installation fassent partie des contraintes à prendre en compte.

7.6 Conclusion du chapitre

Ce chapitre introduit un modèle de programmation linéaire utile pour le positionnement de capteurs dans une pièce. Les caractéristiques des capteurs et la grande portée de détection des nouveaux équipements tendent à rendre moins intéressant le modèle en ce sens que la plupart des pièces ont besoin de deux ou trois capteurs pour être couvertes en totalité. Cependant, le modèle garde toute son utilité parce qu'appliqué au domaine de l'intelligence ambiante. En effet, les exigences du domaine imposent que les informations collectées soient précises et concises sinon on se retrouve dans l'incapacité de mieux raisonner sur l'assistance à offrir.

Le but de ce modèle est d'offrir la possibilité d'ajouter de nouvelles contraintes ou de nouvelles fonctions d'objectif. Chaque assistance étant particulière, il est bien

7.6. CONCLUSION DU CHAPITRE

d'avoir un outil capable de prendre en compte l'expression des besoins donnés. Nous pouvons imaginer une contrainte dans laquelle on exige une observation du lit à 100% par un capteur, ou encore, une seconde contrainte portant sur le fait qu'un des capteurs doit être orienté vers une zone précise. De plus, le modèle permet un raffinement incrémental des contraintes et de la fonction objective à maximiser ou à minimiser.

Cependant, plusieurs exceptions possibles sont probables aux vues de l'orientation des différents capteurs et de la combinaison d'un positionnement au plafond et au mur. Il nous revient dans les travaux futurs de mettre en place un mécanisme pour la gestion de ces exceptions.

Chapitre 8

Expérimentation et Résultats

Ce chapitre présente les résultats de l'expérimentation du modèle proposé.

Une première évaluation du modèle d'architecture pour la sensibilité au contexte a été réalisée durant 42 jours à domicile chez une personne âgée atteinte de la maladie d'Alzheimer. Plusieurs sources de données ont permis de collecter les informations nécessaires à l'évaluation et à l'élaboration des résultats obtenus.

8.1 Démarche expérimentale

Rappelons que l'objectif de l'expérimentation à domicile était triple. En premier, il s'agissait d'accompagner le participant durant ses épisodes d'errance nocturne pour qu'il retourne se coucher de manière sécuritaire. Le deuxième objectif était d'obtenir des informations permettant de structurer les connaissances pour construire l'ontologie, dans l'optique d'offrir une analyse et un traitement de haut niveau des informations contextuelles liées à l'intelligence ambiante dans des conditions réelles. Le dernier objectif était de tester le modèle architectural de sensibilité au contexte.

8.1.1 Profil de la participante

La participante est de sexe féminin et nous l'appellerons Pauline¹. Pauline vit seule dans un triplex dont elle n'utilise que le rez-de-chaussée. Sa maison est située en

1. Pauline est un nom d'emprunt

8.1. DÉMARCHE EXPÉRIMENTALE

campagne et son proche aidant vient tous les matins et tous les soirs pour s'enquérir de ses nouvelles. Lors de son passage le matin, le proche aidant de Pauline organise la journée de Pauline, lui procure quelques soins et l'aide pour sa toilette. Le soir, elle s'assure que Pauline a passé une bonne journée et l'aide à se mettre au lit. Dans la journée, Pauline reçoit par intermittence la visite de quelques proches. Pauline a un chat et aime jouer à des jeux de puzzle (casse-tête). Elle a 78 ans et travaillait dans l'industrie de la couture.

8.1.2 Protocole expérimental

L'expérimentation menée a été conduite en quatre phases comme le montre le Tableau 8.1. Une fois l'obtention des autorisations par le comité d'éthique du CIUSSS (Centre Intégré Universitaire de Santé et de Services Sociaux) , nous avons procédé au recrutement de participants. Le recrutement était conditionné par un ensemble de critères, entre autres le fait que la personne devait avoir plus de 65 ans, avoir une démence du type Alzheimer, vivre à domicile, ne pas souffrir de symptômes dépressifs sévères ni d'antécédent d'alcoolisme... Des séances de travail avec le proche aidant ont eu lieu pour établir les scénarios d'assistance, les besoins et les exigences de Pauline. Nous lui avons demandé l'impact de la maladie de Pauline sur sa journée et sa nuit. D'autres questions ont permis d'élaborer le quotidien de Pauline, ses intérêts et ses occupations régulières. En conclusion, grâce à ces séances nous avons conçu le processus de transformation de l'habitat en habitat intelligent pour faire de la collecte d'information ambiante. Il est cependant, important de noter que cette collecte a évolué durant l'expérimentation.

La première visite chez Pauline avait un double objectif. L'expérimentation a débuté par la passation des tests cognitifs DRS (Dementia Rating Scale) et MoCA (Montréal Cognitive Assessment) pour évaluer les capacités et déficits cognitifs de la participante. Puis nous avons recueilli les signatures pour les formulaires de consentements. En suite, la visite de l'appartement a été faite en présence du proche aidant et de Pauline. Cette visite a permis d'établir une cartographie de l'espace impliqué dans les scénarios d'assistance (la SMARTDOMUS). Ce processus permet de rendre facile l'identification des éléments sur lesquels seront posés les capteurs.

8.1. DÉMARCHE EXPÉRIMENTALE

La deuxième visite consistait à installer les capteurs. Environ, une trentaine de capteurs sans fil ont été installés, comme précisé dans la section 8.3. Nous avons suggéré à Pauline de continuer à vivre comme d'habitude sans se soucier du matériel déployé dans sa maison.

Durant la première semaine nous avons collecté, éprouvé et validé les données pour nous assurer du bon fonctionnement des dispositifs de collecte. Durant la semaine de vérification, le scénario a été ajusté aux habitudes de vie de Pauline et la collecte a été raffinée par l'ajout de nouveaux capteurs. Puis nous avons ajusté quelques capteurs en changeant leur orientation ou en les déplaçant.

La première phase de l'expérimentation a duré 14 jours. Durant ces 14 jours, aucune assistance n'était offerte. Les données étaient récoltées sur les activités de Pauline. Chaque donnée collectée était corroborée par les renseignements obtenus auprès de Pauline et de son proche aidant. Ces données ont servi de seuil et ont permis de définir l'assistance à offrir. Parmi les données collectées, nous avons celles concernant l'activité de Pauline durant son sommeil. Une montre actigraphe portée par Pauline durant toute l'expérimentation mesurait l'activité du sommeil de manière prolongée. Cette montre possédait des capteurs d'accéléromètre, de luminosité, de température, de tremblement parkinsonien, etc. Il nous retournait les informations sur le temps d'immobilité pendant le sommeil et l'heure des éveils, les mouvements durant le sommeil, le temps passé éveillé et la fragmentation du sommeil.

Une deuxième installation faite suite aux 14 jours de collecte de données a permis d'installer des effecteurs/actionneurs dans la maison pour accompagner Pauline en cas d'errance nocturne. Cette installation a duré trois jours afin de tester et ajuster les effecteurs aux besoins de Pauline. Cinq effecteurs avaient été disposés dans son appartement. Une lampe à l'entrée de sa chambre, une lampe dans sa cuisine, une lampe au salon et un haut-parleur pouvant jouer deux types de messages pré-enregistrés différents. Les voix des enregistrements audio sont celles du proche aidant. Le premier message audio, celui que nous appelons "activation audio cuisine" est celui qui l'invite à retourner se coucher et à lui dire qu'elle viendra la voir bientôt. Le second message audio est celui qui l'invite à ne pas sortir, car il est tard et qu'elle doit retourner se coucher. Lors de la deuxième phase de l'expérimentation, une assistance a été apportée à Pauline. Cette phase de collecte et d'assistance a duré 15 jours pendant lesquels

8.1. DÉMARCHE EXPÉRIMENTALE

une analyse continue du système a permis de prévenir d'éventuels problèmes tels que le dysfonctionnement, l'absence de batteries dans un capteur, l'amélioration d'un scénario, etc.

L'expérimentation s'est terminée par le retrait des dispositifs électroniques, par la complétion d'un questionnaire par le proche aidant sur l'acceptabilité de la technologie par les parties prenantes et par la réalisation d'un autre test cognitif DRS et MoCA pour évaluer la progression de la maladie.

8.1.3 Calendrier de l'expérimentation

Pendant les 42 jours d'expérimentation répartis sur six semaines environ, deux semaines ont été consacrées aux tests et ajustements des capteurs. En début de semaine, tous les lundis, nous allions chez Pauline pour changer sa montre actigraphe et changer les batteries des capteurs. Cette visite était réalisée l'avant-midi pour éviter d'influencer les résultats de la nuit. Le Tableau 8.1 montre le chronogramme complet des activités et du processus de déroulement de l'expérimentation chez Pauline.

8.1.4 Résultat du test cognitif de Pauline

Le test MoCA (Montréal Cognitive Assessment) administré par une professionnelle de la santé a duré environ 30 à 35 minutes. Les principaux domaines visés étaient l'attention, la concentration, les capacités d'abstraction, les capacités visuoconstructives, les fonctions exécutives, la mémoire, le langage, le calcul . . .

Les résultats du test MoCA donnent après ajustement un résultat de 18/30 points, ce qui indique la présence d'une atteinte cognitive légère. Initialement, le test brut donnait un résultat de 17/30 points, une correction a été effectuée suivant le niveau de scolarité de la personne.

Le deuxième test cognitif est celui indiqué par l'échelle de démence de Mattis ou encore le DRS (Dementia Rating Scale). Ce test permet d'évaluer le fonctionnement cognitif des personnes soupçonnées de démence. Il met l'accent sur l'attention visuelle et auditive, l'initiation, la persévérance, la mémoire. . . Les résultats du test DRS-1 donnent un total de 115 points et celui du test DRS-2 donne un total de 111 points. A la fin de l'expérimentation, les résultats des tests cognitifs n'ont pas évolué.

8.1. DÉMARCHE EXPÉRIMENTALE

Tableau 8.1 – Calendrier de l'expérimentation

Dimanche	Lundi	Mardi	Mercredi	Jeudi	Vendredi	Samedi
	JOUR 0 VISITE 1 Construction du profil	JOUR 1 VISITE 2 Installation capteurs scénario	JOUR 2 Vérification et test	JOUR 3 Vérification et test	JOUR 4 Vérification et test	JOUR 5 Vérification et test
JOUR 6 Vérification et test	JOUR 7 VISITE 3 Phase de surveillance	JOUR 8 Phase de surveillance	JOUR 9 Phase de surveillance	JOUR 10 Phase de surveillance	JOUR 11 Phase de surveillance	JOUR 12 Phase de surveillance
JOUR 13 Phase de surveillance	JOUR 14 VISITE 4 Phase de surveillance	JOUR 15 Phase de surveillance	JOUR 16 Phase de surveillance	JOUR 17 Phase de surveillance	JOUR 18 Phase de surveillance	JOUR 19 Phase de surveillance
JOUR 20 Phase de surveillance	JOUR 21 VISITE 5 Installation effecteurs Test assistance	JOUR 22 Vérification et test	JOUR 23 Vérification et test	JOUR 24 Vérification et test	JOUR 25 Vérification et test	JOUR 26 Vérification et test
JOUR 27 Vérification et test	JOUR 28 VISITE 6 Phase d'assistance	JOUR 29 Phase d'assistance	JOUR 30 Phase d'assistance	JOUR 31 Phase d'assistance	JOUR 32 Phase d'assistance	JOUR 33 Phase d'assistance
JOUR 34 Phase d'assistance	JOUR 35 VISITE 7 Phase d'assistance	JOUR 36 Phase d'assistance	JOUR 37 Phase d'assistance	JOUR 38 Phase d'assistance	JOUR 39 Phase d'assistance	JOUR 40 Phase d'assistance
JOUR 41 Phase d'assistance	JOUR 42 VISITE 8 Fin questionnaires et tests cognitifs					

8.2 De l'habitat à l'habitat intelligent de Pauline

Pour rendre l'habitat de Pauline intelligent, nous nous sommes fixé quatre contraintes. Ces contraintes sont dictées par les exigences liées aux conditions de vie et de santé de Pauline :

1. La solution de mise en place doit être pervasive et contextuelle. C'est-à-dire être capable de nous donner les informations sur les allées et venues de Pauline de manière omniprésente, mais non intrusive. Il s'agit de comprendre sur la base d'informations ce qu'elle fait sans voir ni entendre.
2. Le système peut parler, mais seulement quand c'est nécessaire et s'il y a urgence. Dans cette situation, des messages vocaux des proches sont diffusés pour aider Pauline et la rassurer.
3. En tout temps, le système surveille son statut et nous en informe. Il est indispensable que les fautes du système soient tout de suite connues afin d'éviter tous désagréments et retards dans l'expérimentation.
4. En tout temps, éviter tout inconvénient à la personne en rendant les technologies déployées calmes et non intrusives.

La maison de Pauline est une maison construite sur un triplex. Dans le cadre de cette expérimentation, seul le rez-de-chaussée a été pris en compte, car il contient les pièces de vie de la participante. Celles-ci comprennent une pièce ouverte comprenant le salon qui communique directement avec la cuisine et la salle à manger. Il faut ajouter à cela une chambre à coucher, un rangement, une salle de bain et une toilette.

Cette expérimentation a nécessité 101 capteurs répartis dans 30 appareils électroniques et quatre effecteurs, dont trois lampes "bloom" et un haut-parleur Bluetooth. Un inventaire du mobilier, des pièces et des habitudes de la vie quotidienne de la personne a été à l'origine du choix et du nombre de dispositifs à utiliser. Ce processus nous permet de produire une cartographie indiquant les lieux visités, la disposition du matériel, l'installation des capteurs et des effecteurs. Nous appelons ce contexte «SMARTDOMUS» c'est-à-dire la partie intelligente d'un habitat intelligent qui est soumise à un processus d'assistance continu pour la satisfaction des besoins de l'utilisateur. La Figure 8.1 montre la «SMARTDOMUS» de Pauline. La représentation a été simplifiée pour faire apparaître uniquement les capteurs et effecteurs.

8.2. DE L'HABITAT À L'HABITAT INTELLIGENT DE PAULINE



Figure 8.1 – La SMARTDOMUS de Pauline

8.3. MATÉRIELS ET MÉTHODES

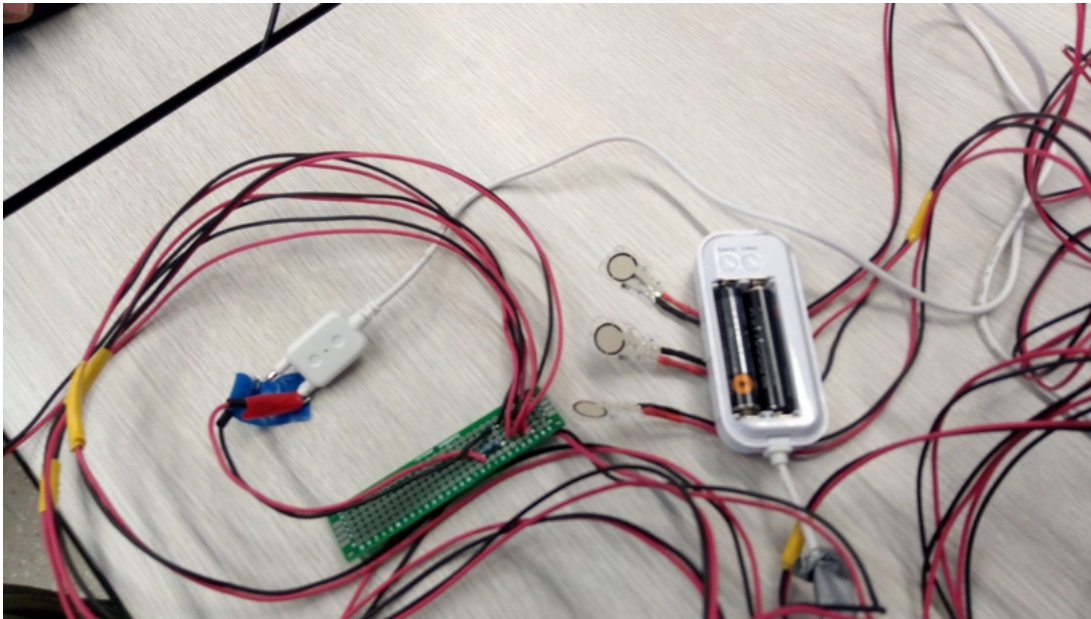


Figure 8.2 – Modification d’un capteur d’eau pour le transformer en capteur de pression

8.3 Matériels et méthodes

L’expérimentation a été adaptée aux exigences de Pauline. Les dispositifs technologiques ont été déployés au fur et à mesure que le scénario se déroulait dans le but de satisfaire au mieux les exigences de Pauline. Le Tableau 8.2 répertorie les périphériques (capteurs, effecteurs, contrôleurs, ordinateurs) utilisés pour l’expérimentation. Toutes les solutions préconisées technologiquement sont des solutions sans fil pour satisfaire certaines contraintes que nous nous sommes fixées en début d’expérimentation (voir section 8.2). Nous avons été appelés à modifier le comportement de certains capteurs filaires pour les transformer en capteurs sans fils. La Figure 8.2 montre une modification du capteur d’eau pour le transformer en capteur de pression afin qu’il soit utilisé sous le lit ou sur les chaises.

8.3. MATÉRIELS ET MÉTHODES

Tableau 8.2 – Liste des équipements utilisés chez Pauline.

Type - Pièce	Dispositif - Action
Chambre	Pression – se coucher, se lever, s’asseoir sur le lit
	Mouvement – se déplacer, bouger, présence dans la chambre
	Contact – Ouverture, fermeture de portes dans la chambre
	Contact – Ouverture, fermeture d’armoire
Toilette	Pression – Toilette
	Mouvement – se déplacer, bouger, présence dans les toilettes
	Contact – Ouverture, fermeture de portes
	Contact – Ouverture, fermeture boîte à pharmacie
	Détecteur d’eau – Ouverture fermeture chasse
	Détecteur d’eau – Ouverture fermeture robinet
Cuisine	Détecteur courant électrique – micro-onde
	Mouvement – se déplacer, bouger, présence dans la cuisine
	Contact – Ouverture, fermeture de portes
	Contact – Ouverture, fermeture armoire
	Contact – Ouverture, fermeture frigo
	Contact – Ouverture, fermeture réfrigérateur
	Contact – Ouverture, fermeture micro-onde
Capteurs salon	Interrupteur électrique – Télévision
	Mouvement – se déplacer, bouger, présence dans le salon
	Contact – Ouverture, fermeture de portes
	Contact – Ouverture, fermeture armoire
	Pression – se lever, s’asseoir
	Pression – se coucher, se lever, s’asseoir
Capteurs externes	Mouvement – se déplacer, bouger, présence à la véranda
	Mouvement – se déplacer, bouger, présence sur les escaliers
Capteurs ambiants	Température, Humidité, Éclairage
Effecteurs	Lampe Philips Hue, signal lumineux
	Hautparleur Bluetooth
Capteur porté	Montre actigraphe – accéléromètre – Microsoft Band 2
Contrôleurs	Vera lite 3, RaZberry, Z-stick
	Arduino
Répéteur	Répéteur Z-wave
Ordinateur	Ordinateur

8.4 Discussion sur les résultats

Cette expérimentation avait un triple objectif. Le premier était la validation du modèle architectural de sensibilité au contexte que nous avons mis en place. Le deuxième était d'offrir à Pauline une assistance durant ses épisodes d'errance nocturne. Le dernier était la collecte de données pour structurer les connaissances autour de la construction de l'ontologie.

8.4.1 Évaluation de l'objectif : Architecture de sensibilité au contexte

L'intérêt d'organiser l'information collectée en couche permet d'agir à différents niveaux pour favoriser des traitements spécifiques ou des structurations particulières afin d'orienter les traitements. Cette démarche permet donc de faire varier en fonction du contexte le comportement du système. Un nouveau contexte ou une nouvelle situation de contexte implique un traitement spécifique.

La suite du document fera ressortir les critères suivants nécessaires à la comparaison de modèles de contexte. Il s'agit :

- a)** De la structuration de l'espace et des informations collectées ;
- b)** De vérifier l'indépendance existant entre les couches ;
- c)** De la validation couche par couche et multi niveaux des informations ;
- d)** De la provenance des observations sous forme d'entités distribuées ;
- e)** De l'encapsulation d'information lors du passage d'une couche à une autre.

De la collecte des informations de contexte

Pour la collecte des informations, nous nous sommes assurés que la SMARTDOMUS était capable en tout temps de donner les informations sur ce que fait la personne et où elle se trouve. Nous disposons d'un module d'acquisition des données :

- Contexte d'acquisition qui collecte trois catégories d'informations. La première catégorie est celle des informations de déplacements et de présence. Ces informations sont fournies par des capteurs de mouvement, de pression et les tapis

8.4. DISCUSSION SUR LES RÉSULTATS

tactiles. La deuxième catégorie est celle des informations d'actions collectées par des capteurs de contact électromagnétiques, d'écoulement d'eau et de mesure de la quantité de courant électrique. La dernière est celle des informations ambiantes en utilisant des capteurs de luminosité, de température, d'humidité et de vibration.

- Contexte de filtrage : le filtrage est mis en place par un mécanisme de gestion de l'hétérogénéité des données grâce au logiciel libre et gratuit OpenHab. Il permet de faire abstraction des différents protocoles de communication utilisés par les contrôleurs pour retourner un format de données uniformisé et standardisé.
- Le contexte d'abstraction mis en place pour séparer les données de maintenance des capteurs de ceux de leur acquisition. Nous avons développé trois interfaces d'abstraction : la première interface fournissant les données sur le niveau de batteries des capteurs ; la seconde fournissant les informations de maintenance et de gestion du réseau de capteurs. La troisième interface est celle permettant la restitution des informations collectées.
- La modélisation du scénario utilisateur a été réalisée avec l'ensemble des membres de l'équipe projet, puis raffinée avec la personne et son proche aidant durant toute l'expérimentation.
- La modélisation du contexte utilisateur permet de mieux personnaliser les réponses du système. Cette modélisation présente l'utilisateur et ses comportements.

De la détection des changements de patrons de contexte

Dans le module de traitement, nous avons utilisé un système basé sur les règles. Le langage Drools utilisé pour la circonstance nous a permis pour chaque patron de contexte de sélectionner l'action à exécuter. Les éléments constituant les patrons de contexte sont décrits comme suit :

- le contexte de composition combine les valeurs de capteurs dans le module de gestion de règles de OpenHab pour produire des patrons de contexte répondant au changement de situation.

8.4. DISCUSSION SUR LES RÉSULTATS

Tableau 8.3 – Caractéristique de Pauline.

Identité :	Pauline
Adresse	vit seule dans une campagne dans un duplex, mais occupe uniquement le rez-de-chaussée.
Données démographiques :	78 ans, Femme, veuve, formation secondaire, mère de plusieurs enfants
Activités professionnelles :	anciennement dans l'industrie de la couture. Maintenant, ne fait plus rien.
Activités domestiques et loisirs	Passionnée des jeux de puzzle, participe à des activités de cuisines collectives, regarde beaucoup la télévision.
Buts	avoir un meilleur cycle circadien
Connaissances et expérience des technologies	aucune
attitude à l'égard des technologies	pas d'opinions,
Maladies associées	Diabète, hypertension, démence du type Alzheimer modérée
Capacités perceptuelles	utilise les lunettes juste pour la lecture uniquement
Capacités physiques	se déplace sans aide et garde une bonne condition physique
Communication	utilise un téléphone analogique, reçoit par intermittence de la visite,

8.4. DISCUSSION SUR LES RÉSULTATS

- Nous sommes passés d'un modèle clé-valeur à un modèle patron-action. Les dix patrons de contextes qui ont été construits sont : "Présence au lit", "présence dans la chambre", "présence dans le salon", "présence dans la cuisine", "présence dans la salle de bain", "sortir de l'appartement", "présence sur la chaise berçante", "prendre son repas", "faire la vaisselle" et "regarder son émission"
- Le contexte de médiation implémente le patron de conception médiateur pour gérer l'organisation des patrons de contextes. Chaque patron avait la même priorité, vu que le profil de Pauline n'indiquait rien de contraignant à ce sujet. Il a été particulièrement utile pour la gestion des alertes audio, lorsque Pauline est sortie de chez elle.

Adaptation et modification dynamique du comportement de l'application

Nous avons mis en place la boucle de rétroaction en implémentant le patron de conception adaptateur et proxy. Le proxy était utilisé pour savoir s'il était nécessaire de prendre en compte un nouveau contexte, une nouvelle ressource ou un nouveau service. Dans notre cas la ressource était un fichier JSON contenant la liste et les caractéristiques de l'ensemble des terminaux nécessaires à fournir l'aide à l'utilisateur. Un service externe est chargé de mettre à jour le fichier en cas de non-réponse, de dysfonctionnement ou de simple arrêt du terminal. Nous n'avions qu'un seul terminal disponible pour notre expérimentation le haut-parleur Bluetooth. Le patron de conception décorateur nous permettait d'ajuster dynamiquement le contexte chaque fois que celui-ci était mis à jour.

Finalement, ces informations sont envoyées au contexte de raisonnement qui, comme nous l'avons illustré plus haut, utilise la logique de premier ordre pour inférer sur les patrons de contextes. L'exécution de la règle déclenche immédiatement l'action associée à celle-ci. Le rôle du gestionnaire de contexte se réduisait au fonctionnement de l'ensemble tout en appelant un service de persistance des informations qui sauvegarde les informations dans une base de données. Il a aussi été utilisé pour faire fonctionner un micro-service de fiabilité du système. Ce mécanisme de gestion des fautes et d'erreurs fonctionnait sous forme de "Ping" et "Pong". Le programme de "Ping" lisait les informations de la sortie de l'intergiciel et l'état des batteries, puis envoyait les informations collectées vers un fichier. Le programme "Pong" lisait ce fichier

8.4. DISCUSSION SUR LES RÉSULTATS

toutes les 15 secondes et comparait son contenu à un historique de lecture qu’il avait créé lors de sa dernière opération. En cas de soucis (égalité dans l’historique, batterie à 28% déterminer par une étude empirique), un courriel était envoyé qui décrivait la nature des soucis. Le même programme envoyait à intervalles réguliers de 15 minutes des messages de battements de cœurs pour savoir si le système fonctionnait bien.

8.4.2 Évaluation de l’objectif : de collecte de données

Durant six semaines, le système a collecté des données en continu 7 jours/7 et 24 heures/24. Les données relatives aux jours de tests après installation des capteurs et installation des effecteurs n’ont pas été supprimées de l’ensemble des données. Un problème de stabilité des capteurs fixés au mur et une panne d’énergie électrique nous ont empêchés durant deux jours de collecter les données. Puis un dysfonctionnement du système et la non-disponibilité des habitants ont fait que nous nous retrouvions à trois jours sans prise de données. Comme nous nous intéressions essentiellement aux données collectées durant la nuit, une restriction a été appliquée à l’ensemble des données collectées. Seules les données entre 20 heures et 8 heures du matin ont été retenues.

Au total, 1 827 789 entrées ont été enregistrées dans la base de données. Après le processus d’étiquetage et de nettoyage (suppression de bruits), des données de synchronisation et des données réseau, 1 523 656 entrées ont été retenues. Puis la restriction sur les données concernant la nuit a donné un total de 360 174 entrées.

Dans son mémoire de maîtrise au sein du laboratoire DOMUS, Robert Radziszewski [87] a montré que les données collectées par les capteurs étaient tout aussi fiables que les données de l’actigraphe (issue des capteurs de la montre portée au bracelet). En effet, durant une expérimentation à domicile chacune des données collectées par les capteurs a été comparée aux informations de l’actigraphe et de l’agenda du sommeil (journal où la personne renseigne les informations sur sa nuit). Il en ressort une corroboration des différentes sources de données. Pour cette raison, nous n’allons pas tenir compte pour la suite des données de l’actigraphe.

Dans cette étude, nous allons mettre l’accent sur le comportement des nuits de Pauline. La Figure 8.3 indique la durée moyenne de présence de Mme Pauline par

8.4. DISCUSSION SUR LES RÉSULTATS

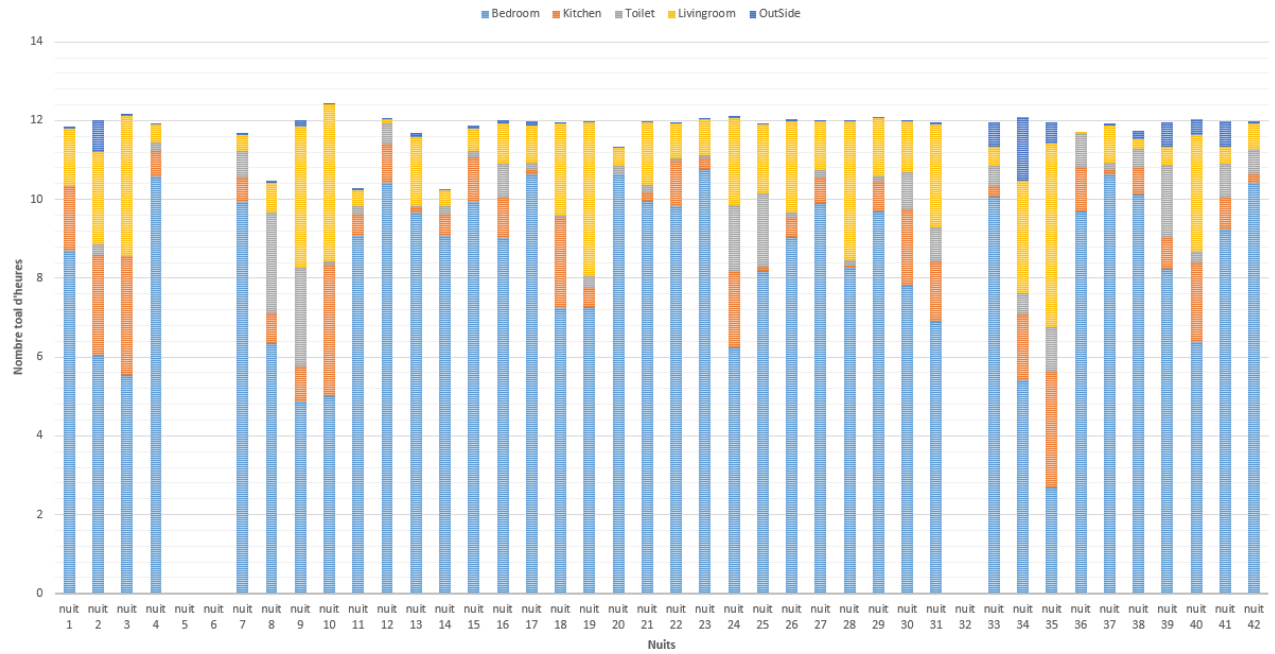


Figure 8.3 – Total du temps passé par nuit et par pièce entre 20 heures et 8 heures.

pièce pour chaque nuit.

Les bandes bleu clair indiquent la chambre, celles en oranges la cuisine, les jaunes le salon et les bleus foncés l'extérieure de la maison.

La Figure 8.3 montre que, durant la nuit, Pauline a passé plus de temps dans sa chambre que dans les autres pièces. Les agitations remarquées lors des premières nuits auraient été causées par notre présence, au vu de l'ensemble de la technologie déployée.

La Figure 8.4 affiche le nombre de fois qu'elle est sortie de sa chambre et le temps passé hors de la chambre à coucher. Le compteur commence au moment où Pauline se couche pour la première fois dans son lit et se lève par la suite pour satisfaire un besoin quelconque. Toutes les sorties sont alors comptabilisées. Nous entendons par "se coucher" le fait de détecter la présence de Pauline sur son lit et nulle part ailleurs dans la maison.

Le graphique de la Figure 8.4 montre que durant les nuits huit et neuf, Pauline a cumulé plus de six heures hors de sa chambre pour une moyenne de 10 sorties. Les

8.4. DISCUSSION SUR LES RÉSULTATS

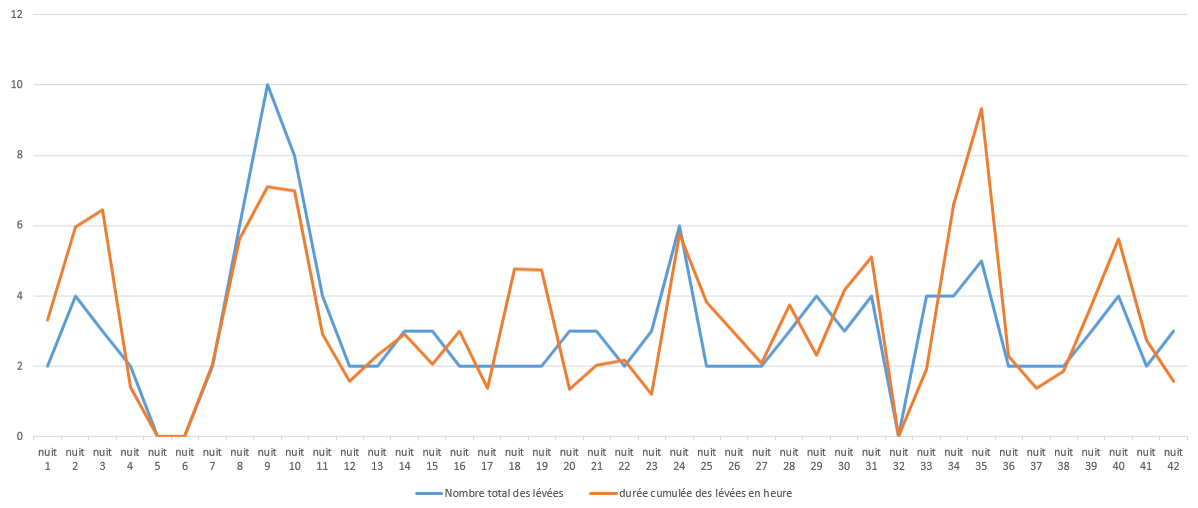


Figure 8.4 – Temps passé hors de la chambre à coucher par nuit.

proches aidantes ont corroboré que Pauline avait été malade durant cette période. Selon la Figure 8.3, Pauline sort de sa chambre un minimum de deux à trois fois par nuit. Cette information était déjà en partie disponible, dans l’agenda du sommeil fourni par Pauline. Il est remarqué que lors de la 35e nuit, Pauline est sortie cinq fois de sa chambre pour un total de neuf heures. Les informations de mouvements du salon et de la pression sur son sofa nous montrent qu’elle y a passé une partie de la nuit.

Les diagrammes de la Figure 8.4 nous amènent à nous poser des questions sur les regroupements possibles de nuits de Pauline. Pour obtenir plus amples d’informations, nous décidons de classer ses nuits. Pour ce faire, nous avons décidé de soumettre les données de la chambre à un algorithme de classification. Pour la circonstance, nous avons utilisé l’algorithme Simple K-means pour mettre en évidence les regroupements intrinsèques des nuits. Cet algorithme est disponible dans le logiciel Weka 3.8.3 [49]. Ce sont les données lissées en les ramenant à l’arrondi de leur borne inférieure qui ont été soumises à Weka pour analyse.

Le moteur de pré traitement de Weka présente une sélection d’attributs ayant neuf classes distinctes (Figure 8.5). La plus grande classe contient 13 instances. Il s’agit de la classe contenant les nuits pour lesquelles Pauline a passé plus de 9 heures dans sa chambre. Puis la classe contenant 9 instances est celle des 10 heures par nuit passée

8.4. DISCUSSION SUR LES RÉSULTATS

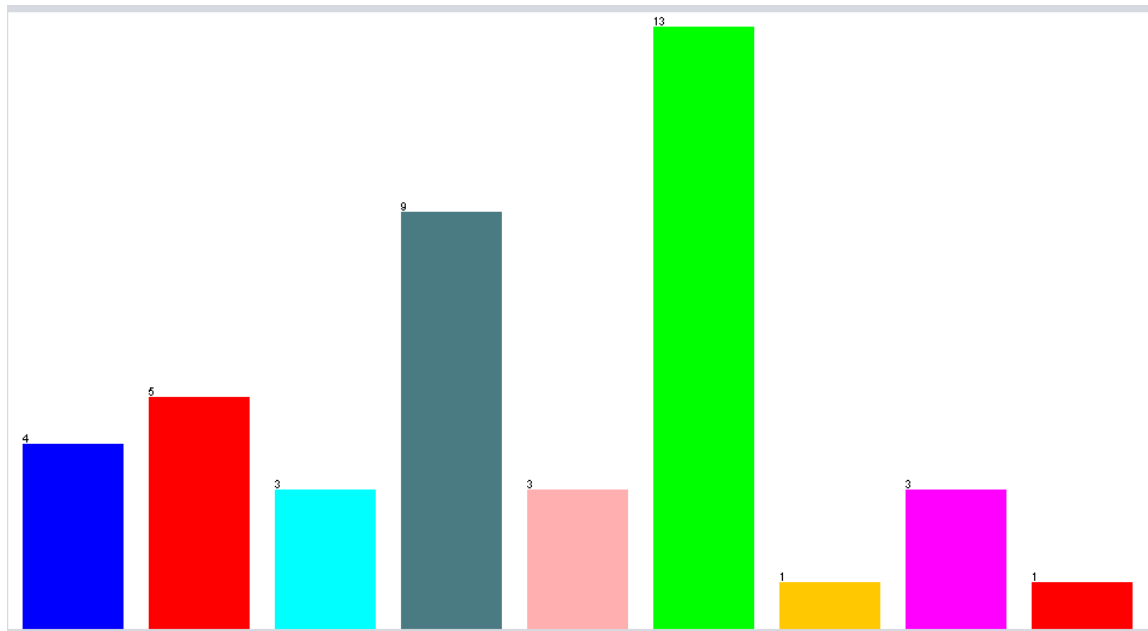


Figure 8.5 – Classification des nuits par regroupement d’heures avec le nombre d’instances par classes

dans la chambre. Pour les classes 1, il s’agit respectivement des nuits à 2 heures et à 4 heures.

De trois à neuf groupes ont été soumis au K-means de Weka. Il en ressort que la meilleure classification est obtenue pour cinq groupes, car elle présente des liens inter groupes forts et des liens extra groupes faibles. Les résultats obtenus sont disponibles au Tableau 8.4.

Le grappe contenant les neuf heures passées dans la chambre est conservé. Il représente le plus grand grappe avec 31% des nuits. Le grappe avec le moins d’instances est celui pour lequel les nuits médianes sont égales à 4 heures. Le résultat de la grappe 0 indiquant 26% des nuits est trompeur. En effet, la grappe prend en compte les trois nuits pour lesquelles le système n’a pas fonctionné.

8.4. DISCUSSION SUR LES RÉSULTATS

Tableau 8.4 – Résultats de la classification des nuits par K-means

	Grappe 0	Grappe 1	Grappe 2	Grappe 3	Grappe 4
Nombre de nuits	11	5	13	4	9
Pourcentage	26%	12%	31%	10%	21%
Heures médianes des grappes	5	6	9	8	10

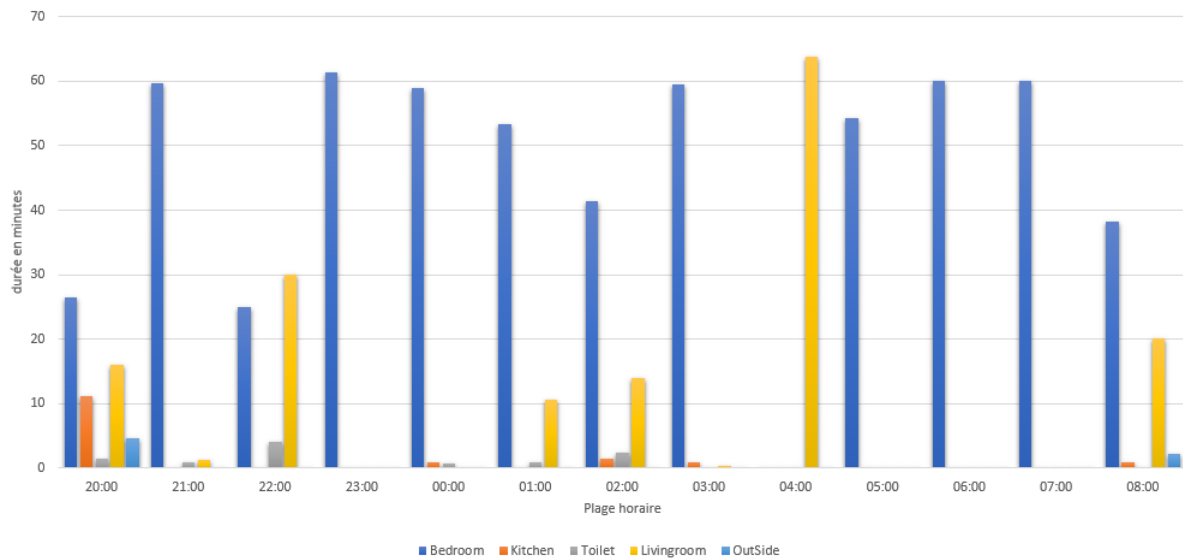


Figure 8.6 – Zoom sur les activités par pièce et par minute pour la 30^e nuit

8.4.3 Évaluation de l'objectif : l'assistance offerte à Mme Pauline

Pour évaluer l'impact de l'assistance à l'errance nocturne offerte à Pauline, nous nous proposons de faire une sélection sur quelques nuits afin de comprendre si Pauline faisait de l'errance nocturne d'un point de vue des capteurs et des effecteurs.

En faisant un zoom sur la 30^e nuit, la Figure 8.6 montre l'organisation des activités en fonction des différentes pièces. Nous constatons qu'à 20 heures Pauline est dans sa chambre puis va à la cuisine, ensuite aux toilettes, au salon avant de revenir dans la chambre. Nous pouvons imaginer que c'est pour satisfaire un besoin aux toilettes que Pauline est sortie du lit. Cependant, c'est après 22 heures qu'elle est restée longtemps sur son lit. On constate aussi que, entre 4 heures et 5 heures du matin, Pauline est sortie de sa chambre pour s'installer sur son sofa.

8.4. DISCUSSION SUR LES RÉSULTATS

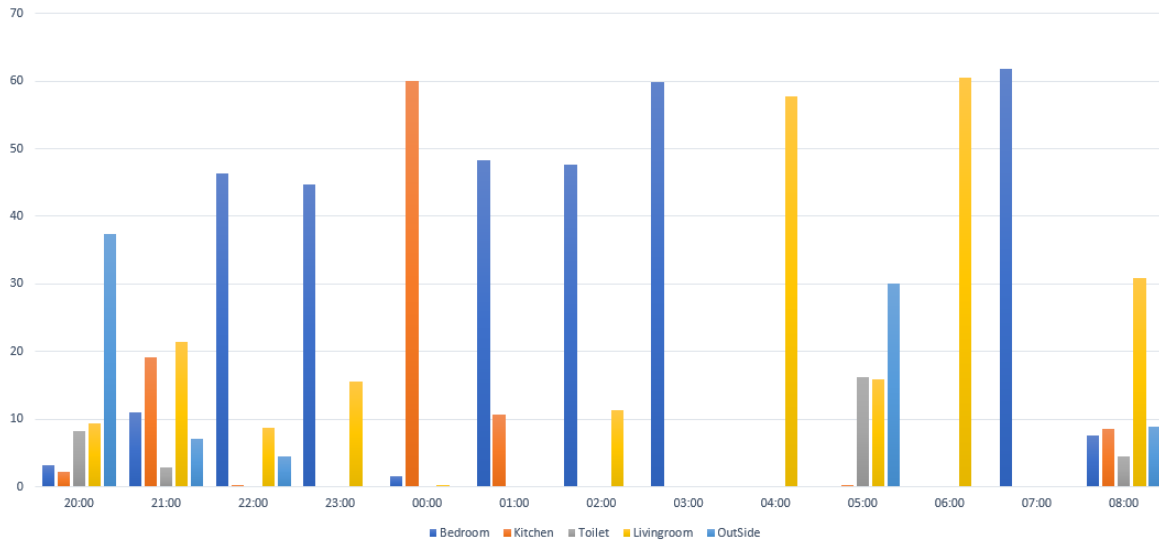


Figure 8.7 – Zoom sur les activités par pièce et par minute pour la 34^e nuit

Si nous étudions la 34^e nuit, on se rend compte que Pauline n’a pas beaucoup dormi et a plutôt eu des épisodes d’errance nocturne. La Figure 8.7 montre qu’elle a fait plusieurs allers et retours entre sa chambre et les autres pièces. Nous constatons qu’à 3 heures elle a quitté sa chambre et a passé le reste de la nuit au salon et c’est seulement à 7 heures qu’elle est retournée dans sa chambre. Ces allers et retours démontrent de l’errance nocturne.

Cependant, Pauline n’a pas que des mauvaises ou des nuits moyennes. La classification a donné un plus haut pourcentage de nuits avec plus de 9 h passées dans la chambre. Un exemple de bonne nuit chez Pauline se présente comme celle illustrée à la Figure 8.8. Dans ce graphe, Pauline est allée se coucher comme toujours à 20 h, mais c’est effectivement à 22 h qu’elle est restée dans sa chambre, et ce jusqu’à 8 h du matin. Notons qu’elle est tout de même allée aux toilettes 3 fois.

Maintenant que nous avons observé trois nuits de Pauline, nous nous proposons de comparer son environnement avec ou sans effecteurs. L’installation des effecteurs s’est faite dès la 21^e nuit. Le graphique de la Figure 8.9 montre une comparaison pièce par pièce des activités de Pauline avec et sans effecteurs dans l’environnement. Il en ressort que Pauline a passé plus de temps dans sa chambre dans l’environnement avec effecteurs. Elle a passé moins de temps dans sa cuisine pour la phase avec effecteurs.

8.4. DISCUSSION SUR LES RÉSULTATS

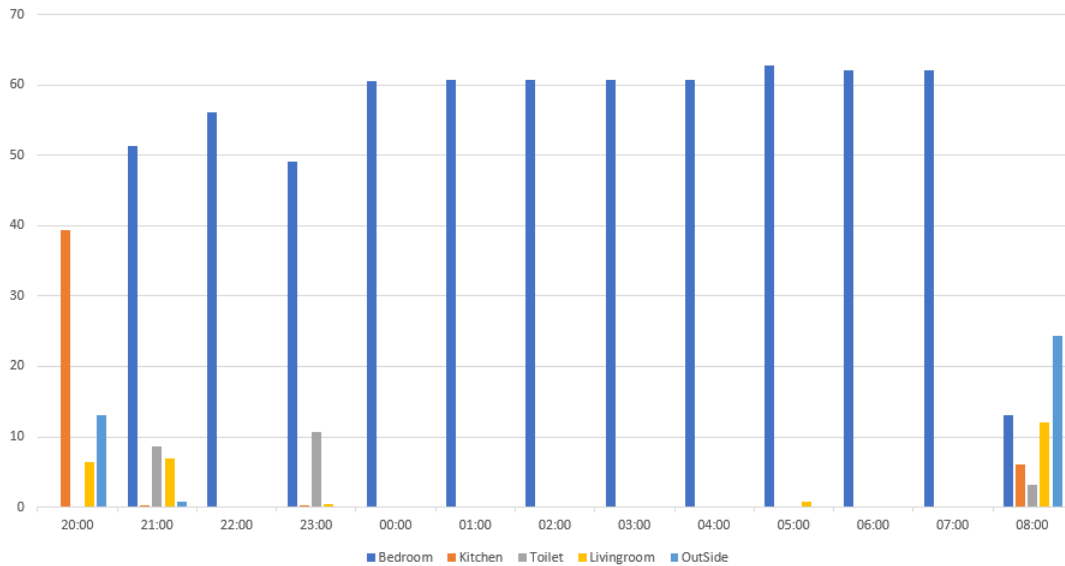


Figure 8.8 – Zoom sur les activités par pièce et par minute pour la 37^e nuit

Il en ressort aussi, qu'elle a passé plus de temps dans son salon. Le croisement de ses données avec les capteurs du salon nous montre qu'elle est restée dans son sofa pendant plusieurs heures. C'est probablement cela qui explique la subtile différence existant entre les activités dans le salon. C'est normal qu'il y ait plus d'activités dans les toilettes et l'extérieur. En effet, Pauline va aux toilettes au moins 2 fois par nuit et tous les matins et les soirs sa proche aidante vient s'occuper d'elle. D'où la récurrence des activités de l'aidant le soir quand elle sort entre 20 h et 21 h et le matin quand elle revient vers 8 h.

Le bleu clair indique la première phase de l'installation celle où il n'y a pas d'effecteurs installés. Les bandes oranges indiquent la seconde phase de l'expérimentation avec effecteurs et capteurs installés dans l'habitat.

Pour mieux comprendre l'impact des effecteurs sur les activités de Pauline, nous avons regardé si l'assistance s'est déclenchée chaque fois qu'elle était requise. La Figure 8.10 présente les activations des effecteurs par rapport aux nombres de levées par nuit. Elle montre que le système a réagi chaque fois que cela était nécessaire. C'est-à-dire quand la personne est en errance depuis plus de 25 minutes. En cas d'errance un rappel est fait toutes les 10 minutes pour l'inciter à retourner se coucher. La nuit du

8.4. DISCUSSION SUR LES RÉSULTATS

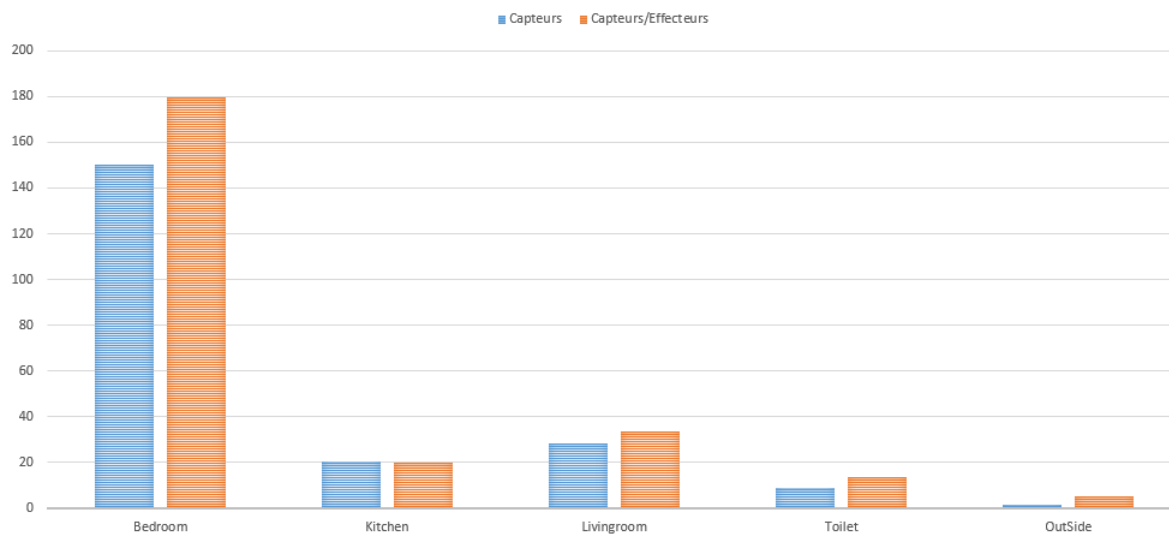


Figure 8.9 – Comparaison des activités dans les pièces avec et sans effecteurs

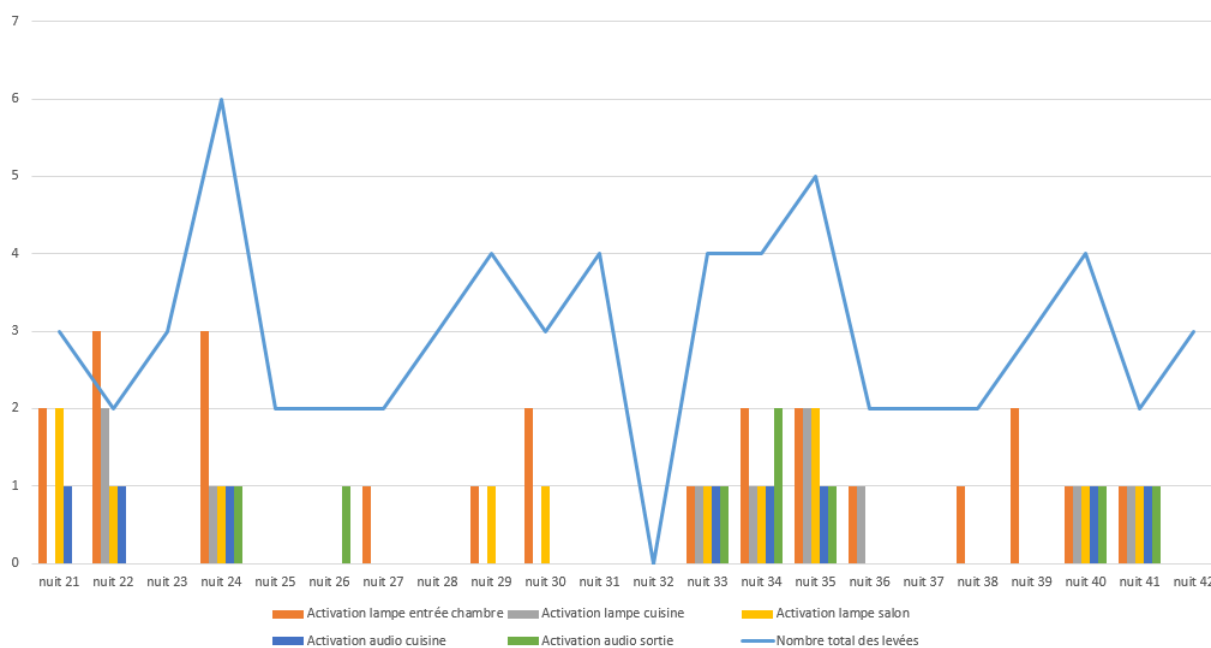


Figure 8.10 – Activation des effecteurs par nuit et par levée

8.5. CONCLUSION

23 montre que Pauline a fait trois sorties et aucune assistance n'a été fournie, parce qu'elle a passé moins de 25 minutes hors de la chambre.

8.5 Conclusion

L'expérimentation à domicile a permis de collecter des informations sur le comportement d'une personne avec une démence du type Alzheimer légère. Elle a surtout permis au proche aidant d'avoir des informations précises sur la qualité de vie de son aidée quand elle n'est pas là. Entre autres, elle a permis de savoir que l'heure de coucher de Pauline était à 22 heures et non à 20 heures comme le croyait son entourage. Nous avons aussi pu constater que l'installation des technologies a eu un impact sur les premières nuits de la participante.

Nos objectifs pour cette expérimentation ont été atteints. Cependant, nous sommes incapables d'évaluer l'impact réel de l'assistance sur les nuits de Pauline. Cela s'explique en partie par le fait que 42 jours d'expérimentations ne sont pas suffisants pour évaluer l'effet de l'assistance sur la réduction des épisodes d'errance nocturne. Nous préconisons pour les prochaines expérimentations de faire au minimum deux mois d'expérimentation avec plus d'un mois d'assistance continue.

Des évaluations et tests sont continuellement menés au sein du laboratoire DOMUS pour raffiner l'organisation architecturale des composants qui forment le modèle. Le socle devant contenir la totalité du modèle sémantique étant encore en cours de développement nous nous proposons de planifier d'autres expérimentations à domicile.

Conclusion

À travers cette thèse, nous abordons la problématique de la prise en compte du contexte dans l'élaboration des systèmes d'intelligence ambiante dans un paradigme DIY. Il a s'agit de produire une cartographie des éléments (capteurs, effecteurs, actionneurs, contrôleurs. . .) de l'Internet des objets nécessaires à la réalisation d'un scénario d'assistance par les personnes ayant peu ou pas de connaissance en informatique. De manière spécifique, il définit les architectures logicielles, les modèles, les spécifications formelles et les connaissances structurelles, pour construire la compréhension ambiante. Le but principal est d'offrir des mécanismes permettant de redonner plus d'autonomie aux aînés ayant ou non des troubles cognitifs. Plus spécifiquement, il propose des approches spécifiant la conception d'un scénario pour produire des composants utiles afin d'empêcher les personnes de programmer/configurer des outils, mais de créer des solutions innovantes. L'expérimentation durant 42 jours à domicile réalisé dans le cadre de ce travail a permis d'évaluer différents aspects de la solution et de proposer des recommandations pour les futures expérimentations. Cette expérimentation visait à réduire l'impact des épisodes d'errance nocturne chez les personnes âgées atteintes d'une démence du type Alzheimer.

Pour aborder cette problématique ainsi que les limites des travaux antérieurs de l'état de l'art, nous avons proposé une approche qui se décline en quatre étapes. La première permet à l'utilisateur dans un processus dirigé par les scénarios de décrire à un niveau d'abstraction assez élevé la manière dont il souhaiterait réaliser ses besoins. Exprimer de façon courante et en langage pseudo-naturel, ces besoins sont le plus souvent ambigus et contradictoires. Notre approche multidisciplinaire utilise donc les spécifications formelles pour valider la cohérence et la conformité des scénarios produits. Le but de cette démarche est de détecter les défauts de conception et non

CONCLUSION

les défauts d'implémentation. Ces spécifications sont soumises à l'outil ALLOY qui génère les instances d'objets orientés modèles pour déterminer les contre-exemples qui violeraient la spécification de base dans une portée donnée. Nous avons opté pour une spécification hiérarchique des scénarios qui s'inspire des modèles d'analyse des tâches. En effet, ils offrent de bons moyens pour décomposer des tâches et ils sont principalement orientés objectifs. Cette première étape du processus permet à l'utilisateur d'exprimer, le quoi et le comment de ses besoins sans ambiguïté.

La seconde étape du processus consiste à structurer les informations dans une base de connaissances. Nous avons utilisé les techniques d'ingénierie des connaissances pour bâtir une ontologie du domaine qui structure la connaissance de l'intelligence ambiante. Le rôle de cette ontologie est d'associer les activités décrites dans le scénario aux dispositifs électroniques, à l'habitat, et au profil de la personne pour produire des connaissances contextuelles capables de rendre le système adaptable et sensible au contexte de l'utilisateur. Notre objectif est de rendre accessibles les connaissances sous une forme définie en fonction du contexte opérationnel. Les solutions actuelles utilisent un raisonnement au niveau de la donnée ou de l'information et sont par le fait même difficilement adaptables lorsque plusieurs entités sont impliquées dans le processus. Le modèle sémantique proposé permet de raisonner à un niveau plus abstrait en utilisant la connaissance plutôt que l'information ou la donnée. Cette étape du processus permet d'organiser la liste des dispositifs nécessaires à la mise en place d'une assistance ambiante d'une part et d'autre part la capacité d'avoir une assistance contextuelle à la réalisation de ses besoins.

La troisième étape est celle de l'organisation architecturale des composants logiciels permettant l'exécution du contexte. Dans cette étape, nous avons utilisé l'approche dirigée par les patrons de conception pour bâtir une architecture générique prenant en compte la spécificité de chaque contexte. Dans le cas de l'errance nocturne, si l'assistance lumineuse est requise alors l'ontologie est capable de proposer diverses catégories de solutions lumineuses possibles puis, suivant le profil de la personne, de la restreindre à une catégorie précise. Organisée en module pouvant s'emboîter grâce à des couches structurelles, l'architecture fonctionnelle utilise des niveaux d'abstraction permettant d'inférer sur la connaissance contextuelle sans tenir compte des spécificités des couches basses.

CONCLUSION

Le travail s’achève par la proposition d’un modèle de programmation linéaire pour le positionnement des capteurs dans l’habitat intelligent. Nous utilisons donc la connaissance disponible dans l’ontologie pour indiquer à l’utilisateur l’emplacement où il devrait installer les capteurs, en lui fournissant ainsi la cartographie des instructions à suivre afin de construire le système d’assistance.

Enfin, un travail important d’expérimentation a été fait pour évaluer certains aspects de notre modèle sémantique. L’expérimentation a suivi un protocole bien défini et validé par un comité d’éthique. La méthodologie a nécessité la mise en place d’un ensemble de scénarios d’expérimentation, du matériel connecté, des tests cognitifs, d’un suivi en temps réel des activités réalisées par la personne. L’analyse des résultats montre que le système fonctionne bien et que nous sommes capables d’accompagner, d’assister et de guider une personne atteinte d’une maladie d’Alzheimer durant des épisodes d’errance nocturne.

Contributions

Ce travail de thèse a permis d’apporter des contributions aussi bien dans le domaine de la recherche fondamentale que dans l’ingénierie logicielle et de l’Internet des objets. La modélisation sémantique pour une approche DIY de la sensibilité au contexte proposée montre qu’il est possible de partir d’une représentation de scénarios non formelle pour aboutir à une spécification formelle de scénarios. Chaque instance de spécifications produites a permis de dégager une sémantique bien définie pour la structuration des différentes connaissances dans une ontologie. Une autre contribution est la mise en place du cadre formel permettant de transformer l’approche classique de raisonnement sur le contexte basé sur un événement une action par un processus de raisonnement sur basé sur des patrons d’événements pour une action.

Dans l’ingénierie des connaissances, nous proposons une ontologie qui permet de spécifier de manière explicite les concepts et les relations existant entre l’environnement (l’habitat, mobilier, pièces, éléments de la maison, etc.), les équipements (capteurs, effecteurs, contrôleurs, etc.), le profil utilisateur et les scénarios (tâches, activités, actions, etc . . .). De manière logicielle nous proposons une démarche architecturale orientée couche permettant d’exploiter l’expérience et les leçons tirées du

CONCLUSION

concept DIY. Ce que nous apportons fondamentalement est la réutilisation sous forme de modèle d'architecture d'expériences pour enrichir les nouveaux besoins, plutôt que d'utiliser le même programme d'assistance et changer les paramètres. L'idée que nous soutenons est que quelqu'un a sûrement déjà résolu le même problème d'assistance donc il est plus intuitif d'utiliser la connaissance qu'il veut bien mettre à disposition. Une autre contribution est la proposition d'un modèle de programmation linéaire pour le positionnement de capteurs dans un habitat intelligent.

Critique du travail

Plusieurs éléments permettent d'évaluer nos travaux. Le premier point d'arrêt de la critique des travaux concerne la réalisation des objectifs de la thèse. Le deuxième concerne les publications scientifiques réalisées pour accompagner ses travaux. Nous terminerons cette partie de critique par la présentation des limites de notre travail.

Satisfaction des objectifs

La revue de la littérature a permis de dégager les besoins et les exigences spécifiques nécessaires à la mise en place d'un écosystème pour la création de solutions innovantes par les utilisateurs finaux. L'objectif est atteint par définition de nouveaux paradigmes et la proposition de nouveaux modèles de structuration de la connaissance et des modèles de positionnement de capteurs. Il permet de faciliter la mise en place de composants utiles aux utilisateurs finaux.

La formalisation des scénarios a permis de satisfaire les besoins sur la création et la description des scénarios. Chaque spécification faite permet de puiser dans la base de connaissances l'intelligence nécessaire pour donner le nombre de capteurs nécessaires à l'assistance ambiante à offrir. L'évaluation de la satisfaction du dernier objectif se matérialise par l'intégration continue et l'utilisation du modèle sémantique dans l'ensemble du kit d'assistance. Les résultats de l'expérimentation à domicile permettent d'évaluer d'une certaine manière certains aspects du modèle.

Plusieurs publications scientifiques ont été réalisées dans le cadre de nos travaux de thèse. Nous pouvons citer entre autres un article [89] dans une revue avec comité

CONCLUSION

de lecture, trois articles de revue de type-conférence internationale avec comité de lecture et publication des actes [88, 65, 80] et plusieurs affiches et des résumés dans des conférences internationales [86, 58, 52]. Enfin, ces travaux ont permis la contribution à un chapitre dans un livre [18].

Limites du travail

Le modèle sémantique d’intelligence ambiante proposé dans cette thèse comporte certaines limitations.

L’une des premières limitations est la non-standardisation de la nomenclature et l’absence de l’utilisation des patrons de conception pour la construction et la structuration de l’ontologie. Nous aurions voulu que notre architecture ontologique soit basée sur des modèles. En effet, elle aurait permis de rendre l’ontologie disponible sous forme de module de conception de contenu d’assistance ambiante DIY. Ceci aurait pour avantage de construire l’ensemble sous forme de méta propriété, de méta classe et non des tentatives comme nous l’avons fait de capturer la nature intrinsèque du paradigme DIY. Beaucoup trop de descriptions que nous proposons sont dépendantes de la perception, des facteurs culturels et des conventions sociales que nous avions.

Une autre limitation majeure est celle survenue pendant la spécification des scénarios. En effet, durant le processus de création d’un nouveau scénario par combinaison de scénarios existants nous avons opté pour la copie complète du sous-arbre plutôt que de créer des références. Ce choix s’est imposé parce que l’utilisation des pointeurs rendrait plus complexe la navigation entre les sous-tâches d’une tâche. De plus, il entraînait en contradiction avec la spécification de base qui stipulait qu’un nœud ne devrait avoir qu’un seul nœud parent.

Travaux futurs de recherche

Les extensions possibles viseraient la définition de procédés pour améliorer le modèle sémantique. Le premier travail à réaliser est celui de la mise en place du cadre de raisonnement sémantique pour assister la personne dans la réalisation des AVQ.

CONCLUSION

C'est-à-dire mettre en place des mécanismes de raisonnement basés sur le langage SWRL plutôt que sur les relations de subsomption, de raisonnement transitif, de chaînage avant et arrière comme c'est le cas actuellement.

Le deuxième point concerne la mise en place d'un dépôt pour la persistance temps réel des informations dans l'ontologie afin de tester le comportement de l'API. En utilisant le dépôt, RDF Stardog 3.0 Marc Chevalaz [25] a fait une étude sur le temps nécessaire pour sauvegarder les notifications des événements dans une ontologie suite à un changement de valeur de capteurs dans un habitat intelligent. Donc la mécanique est possible et a déjà été mise en place, il faudrait étendre les implémentations pour notifier au gestionnaire de contexte les changements de situation (habitat, panne de capteurs, assistance, réalisation ou pas d'AVQ...) survenus et de les stocker dans l'ontologie. Ce processus permettrait de raffiner le raisonnement proposé pour que l'application soit plus sensible au contexte.

Le troisième axe est la conception d'outils de programmation graphiques et interactifs, permettant aux utilisateurs finaux ou tout autres partis prenants de créer des scénarios. Puisque, la validation se fait de manière automatiquement et la cohérence assurée chaque fois qu'une description est apportée au scénario, il aurait été judicieux d'offrir une application cliente comme scratch, sur la base du principe ce que tu vois c'est ce que tu auras, pour faciliter le design des scénarios.

Le quatrième axe est celui de la conception d'un simulateur graphique pour tester différentes instances de solutions du modèle pour le placement des capteurs avec orientation. Durant notre modélisation, plusieurs solutions possibles ont été disponibles suivant l'ajustement d'un ou plusieurs paramètres. Il aurait été bien plus élégant d'avoir une interface graphique qui réagit à chaque ajustement de paramètres.

Perspective

Le domaine de l'intelligence ambiante appliqué à l'assistance aux AVQ pour les personnes âgées en perte d'autonomie ou pas, nécessite la mise en place de méthodologie de recherche pluridisciplinaire et multidisciplinaire. Cette thèse propose un cadre pour prendre en compte les compétences multiples et diverses du domaine de l'assistance afin de les structurer dans une ontologie. Ces compétences et ces exper-

CONCLUSION

tises sont pour la plupart regroupées dans un domaine de multidisciplinarité lorsque les parties prenantes sont associées au processus. La multidisciplinarité permettrait d'utiliser plusieurs disciplines autour du thème de la reprise d'autonomie, alors, que la pluridisciplinarité offrirait à cette juxtaposition de disciplines autour du même type de relations plus de services sensibles au contexte. Concrètement, il s'agirait d'utiliser les connaissances d'ergonomes, d'ergothérapeutes, de gérontologues, d'informaticiens, de psychologues, de professionnels de la santé... pour les associer au domaine de l'informatique (intelligence artificielle, génie logiciel, satisfaction de contrainte, optimisation, spécification formelle, programmation linéaire...) afin d'offrir aux utilisateurs finaux, les capacités de construire eux-mêmes leurs propres solutions d'assistance, afin de redonner l'autonomie à leur proche ou à eux-mêmes.

Annexe A

Les 8 principes de la technologie calme

1. La technologie devrait exiger le minimum d'attention possible :
 - (a) La technologie peut communiquer, mais n'a pas besoin de parler.
 - (b) La technologie crée une conscience ambiante à travers différents sens.
 - (c) La technologie communique des informations sans sortir de la portée de son environnement ou de sa tâche.
2. La technologie devrait informer et créer le calme pour que :
 - (a) La tâche principale d'une personne ne soit pas l'informatique, mais l'être humain.
 - (b) Les gens aient ce dont ils ont besoin pour résoudre leur problème, et rien de plus.
3. La technologie devrait utiliser la périphérie :
 - (a) Une technologie calme se déplacera facilement de la périphérie de notre attention, du centre vers l'arrière.
 - (b) La périphérie informe sans surcharger l'attention.
4. La technologie devrait amplifier le meilleur de la technologie et le meilleur de l'humanité :
 - (a) La technologie est construite pour les gens d'abord.

- (b) Les machines ne doivent pas agir comme des humains.
 - (c) Les humains ne devraient pas agir comme des machines.
 - (d) La technologie amplifie la meilleure partie de chacun.
5. La technologie peut communiquer, mais n'a pas besoin de parler :
 - (a) Le produit doit-il s'appuyer sur la voix ou peut-il utiliser une méthode de communication différente ?
 - (b) La technologie comment communique-t-elle ses changements ?
 6. La technologie devrait fonctionner même en cas d'échec :
 - (a) Pensez à ce qui se passe si votre technologie échoue.
 - (b) A-t-il par défaut un état utilisable ou tombe-t-il complètement en panne ?
 7. La bonne quantité de technologie est le minimum nécessaire pour résoudre le problème :
 - (a) Quelle est la quantité minimale de technologie nécessaire pour résoudre le problème ?
 - (b) Comment affiner la fonctionnalité définie pour que le produit fasse ce qu'il doit faire et pas plus.
 8. La technologie doit respecter les normes sociales :
 - (a) La technologie prend du temps à s'ancrer dans l'humanité.
 - (b) La technologie pourrait violer quelles normes sociales ou causer du stress ?
 - (c) La technologie doit introduire lentement des fonctionnalités pour que les utilisateurs aient le temps de s'habituer au produit.

Annexe B

Manifeste DIY pour L'Internet des objets

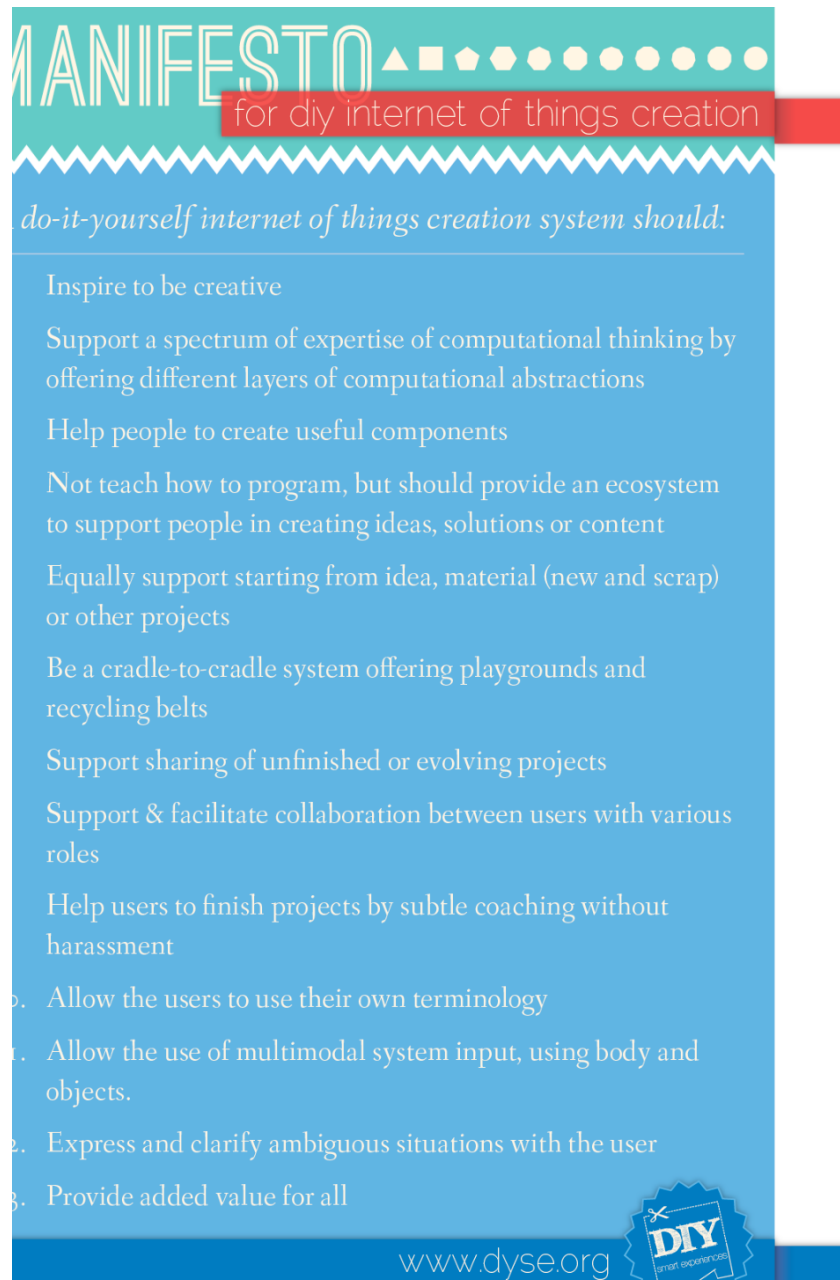


Figure B.1 – Manifeste DIY pour L’Internet des objets [31]

Annexe C

Les instances d'exécution en
ALLOY du scénario errance
nocturne

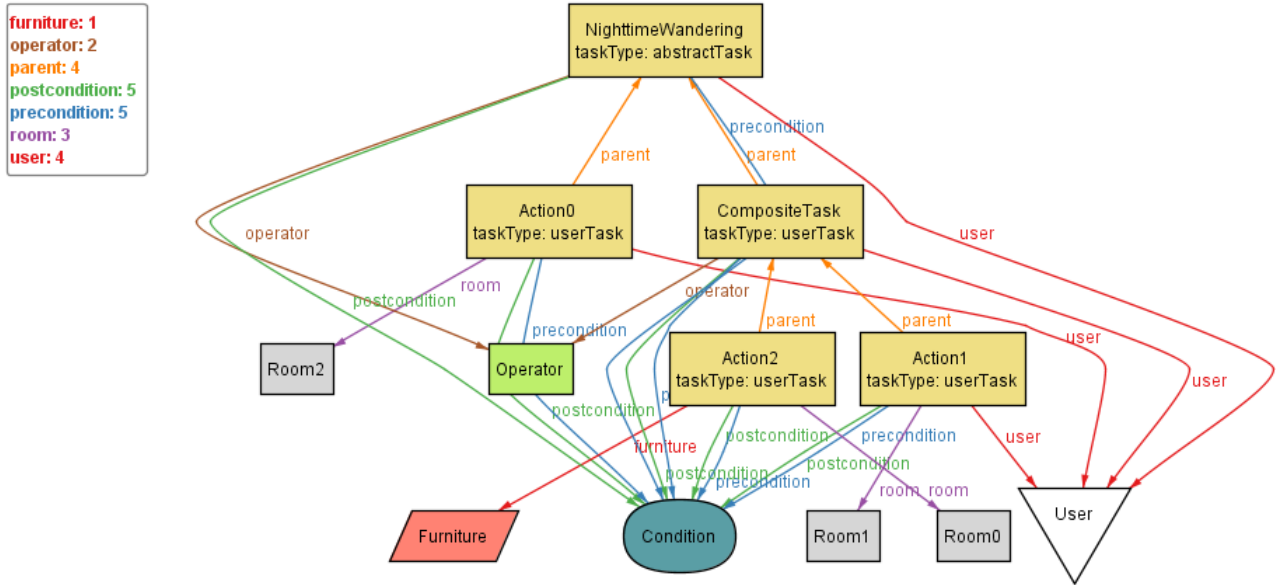


Figure C.1 – Exemple d’instances pour la première itération du scénario errance nocturne

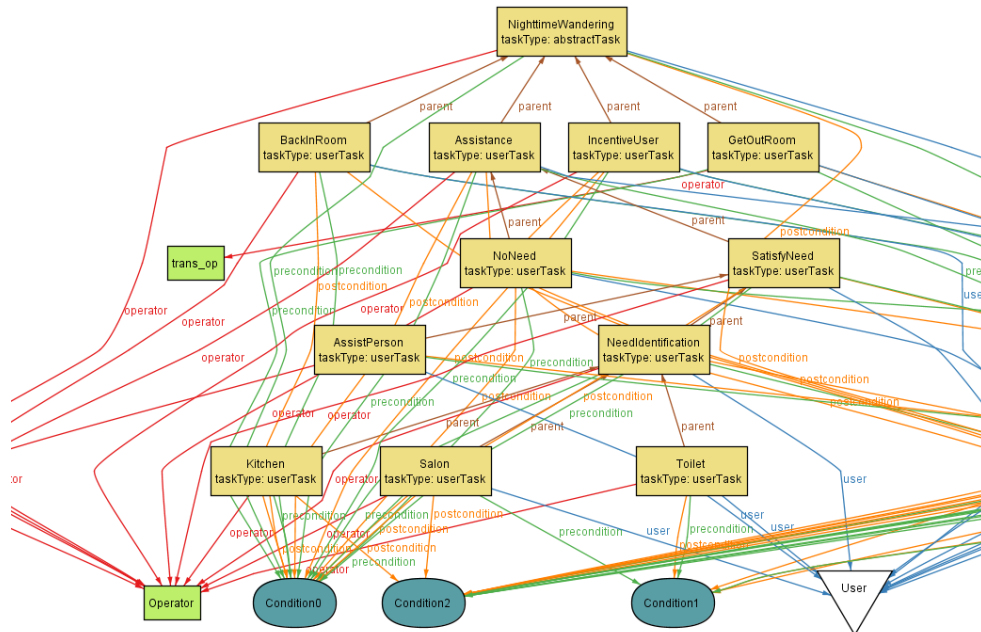


Figure C.2 – Exemple d’instances pour une ième itération du scénario errance nocturne

Bibliographie

- [1] Emillie Emile AARTS et José ENCARNACÃO.
True visions : The emergence of ambient intelligence.
Springer Heidelberg, 2006.
- [2] Gregory D ABOWD, Anind K DEY, Peter J BROWN, Nigel DAVIES, Mark SMITH
et Pete STEGGLES.
« Towards a better understanding of context and context-awareness. ».
In International Symposium on Handheld and Ubiquitous Computing, pages
304–307, 1999.
- [3] Jean-Raymond ABRIAL et Jean-Raymond ABRIAL.
The B-book : assigning programs to meanings.
Cambridge University Press, 2005.
- [4] Jean-Raymond ABRIAL et Louis MUSSAT.
« Introducing dynamic constraints in B ».
Dans *International Conference of B Users*, pages 83–128. Springer, 1998.
- [5] Humaid AL SHU’EILI, G ourab Sen GUPTA et Subhas MUKHOPADHYAY.
« Voice recognition based wireless home automation system ».
Dans *Mechatronics (ICOM), 2011 4th International Conference On*, pages 1–6.
IEEE, 2011.
- [6] Muhammad Raisul ALAM, Student MEMBER, Mamun BIN, Ibne REAZ, Mohd
ALAUDDIN et Mohd ALI.
« A Review of Smart Homes — Past , Present , and Future ».
*IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications
and Reviews)*, 42(6):1190–1203, 2012.

BIBLIOGRAPHIE

- [7] Dean ALLEMANG et James HENDLER.
Semantic web for the working ontologist : effective modeling in RDFS and OWL.
Elsevier, 2011.
- [8] James F ALLEN et George FERGUSON.
« Actions and Events in Interval Temporal Logic ».
Dans Oliviero STOCK, éditeur, *Spatial and Temporal Reasoning*, pages 205–245.
Springer Netherlands, 1997.
- [9] Z-Wave ALLIANCE.
« Z-Wave », 2015.
- [10] T N B ANH et S TAN.
« Real-Time Operating Systems for Small Microcontrollers ».
IEEE Micro, 29(5):30–45, 9 2009.
- [11] John ANNETT.
« Hierarchical task analysis ».
Handbook of cognitive task design, 2:17–35, 2003.
- [12] K. ATUKORALA, D. WIJEKOON, M. THARUGASINI, I. PERERA et C. SILVA.
« SmartEye - Integrated solution to home automation, security and monitoring through mobile phones ».
NGMAST 2009 - 3rd International Conference on Next Generation Mobile Applications, Services and Technologies, pages 64–69, 2009.
- [13] Juan AUGUSTO et Paul MCCULLAGH.
« Ambient Intelligence : Concepts and applications ».
Computer Science and Information Systems, 4:1–27, 2007.
- [14] Clark BARRETT et Cesare TINELLI.
« Satisfiability modulo theories ».
Dans *Handbook of Model Checking*, pages 305–343. Springer, 2018.
- [15] A. BIKAKIS, T PATKOS, G ANTONIOU et D PLEXOUSAKIS.
« A Survey of Semantics-Based Approaches for Context Reasoning in Ambient Intelligence ».
Constructing Ambient Intelligence, 11:14–23, 2008.
- [16] Antonis BIKAKIS et Grigoris ANTONIOU.
« Defeasible contextual reasoning with arguments in ambient intelligence ».

BIBLIOGRAPHIE

- IEEE Transactions on Knowledge and Data Engineering*, 22(11):1492–1506, 2010.
- [17] Tommaso BOLOGNESI et Ed BRINKSMA.
« Introduction to the ISO specification language LOTOS ».
Computer Networks and ISDN systems, 14(1):25–59, 1987.
- [18] Kevin BOUCHARD, Jianguo HAO, Bruno BOUCHARD, Sébastien GABOURY, Mohammed Tarik MOUTACALLI, Charles GOUIN-VALLERAND, Hubert Kenfack NGANKAM, Hélène PIGOT et Sylvain GIROUX.
« The Cornerstones of Smart Home Research for Healthcare ».
Dans *Advanced Data Analytics in Health*, pages 185–200. Springer, 2018.
- [19] Oana BUCUR, Philippe BEAUNE, Olivier BOISSIER et OTHERS.
« Representing context in an agent architecture for context-based decision making ».
Dans *Proceedings of the Workshop on Context Representation and Reasoning (CRR'05), Paris, France*, volume 5, 2005.
- [20] Amber CASE.
Calm technology : Principles and patterns for non-intrusive design.
" O'Reilly Media, Inc.", 2015.
- [21] Y CHARLON, W BOURENNANE, F BETTAHAR et E CAMPO.
« Activity monitoring system for elderly in a context of smart home ».
IRBM, 34(1):60–63, 2013.
- [22] Chao CHEN et Diane J COOK.
« Behavior-based Home Energy Prediction ».
8th International Conference on. IEEE, pages 57–63, 2012.
- [23] Harry CHEN, Tim FININ et Anupam JOSHI.
« The SOUPA Ontology for Pervasive Computing ».
Ontologies for Agents : Theory and Experiences, pages 233–258, 2005.
- [24] Harry CHEN, Filip PERICH, Tim FININ et Anupam JOSHI.
« SOUPA : Standard ontology for ubiquitous and pervasive applications ».
Proceedings of MOBIQUITOUS 2004 - 1st Annual International Conference on Mobile and Ubiquitous Systems : Networking and Services, pages 258–267, 2004.

BIBLIOGRAPHIE

- [25] Marc Henri CHEVALAZ.
« *Gestion de contexte dans un habitat intelligent à base d'ontologie : modélisation, implantation et validation* ».
Thèse de doctorat, Université de Sherbrooke, 2016.
- [26] Myung Eun CHO, Mi Jeong KIM et Jeong Tai KIM.
« Design Principles of User Interfaces for the Elderly in Health Smart Homes ».
Dans *10th International Symposium on Sustainable Healthy Buildings*, pages 45–59, 2013.
- [27] Michael COMPTON, Payam BARNAGHI, Luis BERMUDEZ, Raúl GARCÍA-CASTRO, Oscar CORCHO, Simon COX, John GRAYBEAL, Manfred HAUSWIRTH, Cory HENSON, Arthur HERZOG et OTHERS.
« The SSN ontology of the W3C semantic sensor network incubator group ».
Web semantics : science, services and agents on the World Wide Web, 17:25–32, 2012.
- [28] Diane J. COOK, Juan C. AUGUSTO et Vikramaditya R. JAKKULA.
« Ambient intelligence : Technologies, applications, and opportunities ».
Pervasive and Mobile Computing, 5(4):277–298, 2009.
- [29] Diane J COOK et Sajal K DAS.
« How smart are our environments ? An updated look at the state of the art ».
Pervasive and mobile computing, 3(2):53–73, 2007.
- [30] Diane J COOK et Wenzhan SONG.
« Ambient intelligence and wearable computing : Sensors on the body , in the home , and beyond ».
Journal of ambient intelligence and smart environments, 1(2):83–86, 2009.
- [31] Dries DE ROECK, Karin SLEGERS, Johan CRIEL, Marc GODON, Laurence CLAEYS, Katriina KILPI et An JACOBS.
« I would DiYSE for it ! ».
Proceedings of the 7th Nordic Conference on Human-Computer Interaction Making Sense Through Design - NordiCHI '12, page 170, 2012.
- [32] Dries DE ROECK, Karin SLEGERS, Johan CRIEL, Marc GODON, Laurence CLAEYS, Katriina KILPI et An JACOBS.

BIBLIOGRAPHIE

- « I Would DiYSE for It! : A Manifesto for Do-it-yourself Internet-of-things Creation ».
- Dans *Proceedings of the 7th Nordic Conference on Human-Computer Interaction : Making Sense Through Design*, NordiCHI '12, pages 170–179. ACM, 2012.
- [33] Anind K DEY et Gregory D ABOWD.
« A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications ».
Hum.-Comput. Interact, 16:97–166, 2001.
- [34] Anind K DEY, Gregory D ABOWD et Andrew WOOD.
« CyberDesk : a framework for providing self-integrating context-aware services ».
Knowledge-Based Systems, 11(1):3–13, 1998.
- [35] Claudia DIAMANTINI, Alessandro FREDDI, Sauro LONGHI, Domenico POTENA et Emanuele STORTI.
« A goal-oriented, ontology-based methodology to support the design of AAL environments ».
Expert Systems with Applications, 64:117–131, 2016.
- [36] Dan DING, Rory A COOPER, Paul F PASQUINA et Lavinia FICI-PASQUINA.
« Maturitas Sensor technology for smart homes ».
Maturitas, 69(2):131–136, 2011.
- [37] Faiyaz DOCTOR, Hani HAGRAS et Victor CALLAGHAN.
« A Fuzzy Embedded Agent-Based Approach for Realizing Ambient Intelligence in Intelligent Inhabited Environments ».
IEEE Transactions on Systems, Man, and Cybernetics-Part A : Systems and Humans, 35(1):55–65, 2005.
- [38] Edward R. DOUGHERTY, Robert M. HARALICK, Yidong CHEN, Carsten AGERSKOV, Ulrik JACOBI et Poul Henrik SLOTH.
« Estimation of optimal morphological τ -opening parameters based on independent observation of signal and noise pattern spectra ».
Signal Processing, 29(3):265–281, 1992.

BIBLIOGRAPHIE

- [39] K DUCATEL, M BOGDANOWICZ, F SCAPOLO, J LEIJTEN et J-C BURGELMAN.
« ISTAG Scenarios for Ambient Intelligence in 2010 ».
Society, page 58, 2001.
- [40] Emeric FONTAINE.
« *Programmation d'espace intelligent par l'utilisateur final* ».
Thèse de doctorat, Grenoble, 2012.
- [41] Michael FRIEDEWALD, Olivier DA COSTA, Yves PUNIE, Petteri ALAHUHTA et Sirkka HEINONEN.
« Perspectives of ambient intelligence in the home environment ».
Telematics and Informatics, 22(3):221–238, 2005.
- [42] Carl FROLUND, Jonghong JEON, Younkwon LEE, Seungyun LEE et Hyoungh Jun KIM.
« Study for strengthening the ICT DIY ecosystem ».
International Conference on ICT Convergence, pages 269–274, 2014.
- [43] Florian FUCHS, Iris HOCHSTATTER, Michael KRAUSE et Michael BERGER.
« A metamodel approach to context information ».
Dans *null*, pages 8–14. IEEE, 2005.
- [44] Mathieu GALLISSOT.
« *éliser le concept de confort dans un habitat intelligent : du multisensoriel au comportement* Modéliser le concept de confort dans l ' habitat intelligent : du multisensoriel au comportement ».
Thèse de doctorat, Grenoble, 2012.
- [45] Neil M GOLDMAN.
« Task Computing – The Semantic Web Meets Pervasive Computing ».
Lecture Notes in Computer Science, 2870(October):850–865, 2003.
- [46] Thomas R GRUBER.
« A translation approach to portable ontology specifications ».
Knowledge acquisition, 5(2):199–220, 1993.
- [47] Thomas R GRUBER.
« Toward principles for the design of ontologies used for knowledge sharing? ».
International journal of human-computer studies, 43(5-6):907–928, 1995.

BIBLIOGRAPHIE

- [48] Tao GU et Hung Keng PUNG.
« Toward an OSGi-Based Infrastructure for Context-Aware Applications ».
Ieee Pervasive Computing, 3(4):66–74, 2004.
- [49] Mark HALL, Eibe FRANK, Geoffrey HOLMES, Bernhard PFAHRINGER, Peter REUTEMANN et Ian H WITTEN.
« The WEKA data mining software : an update ».
ACM SIGKDD explorations newsletter, 11(1):10–18, 2009.
- [50] Dae-Man HAN et Jae-Hyun LIM.
« Design and implementation of smart home energy management systems based on zigbee ».
IEEE Transactions on Consumer Electronics, 56(3):1417–1425, 2010.
- [51] Andre Mayers HÉLÈNE PIGOT et Sylvain GIROUX.
« The intelligent habitat and everyday life activity support ».
5th international conference on Simulations in Biomedicine, 2003.
- [52] Dominique Lorrain Isabelle Viens Sylvain Giroux HÉLÈNE PIGOT, Hubert Ngankam.
« Smart home to accompany seniors during nighttime disorders ».
CAG, 2018.
- [53] Jean-Guy Meunier Brigitte Kerhervé André Mayers HÉLÈNE PIGOT, Bernard Lefebvre et Sylvain GIROUX.
« The role of intelligent habitats in upholding elders in residence ».
5th international conference on Simulations in Biomedicine, pages 497–506, 2003.
- [54] Sylvain Giroux Bernard Lefebvre Vincent Rialle HÉLÈNE PIGOT, André Mayers et Norbert NOURY.
« Smart house for frail and cognitive impaired elders ».
First International Workshop on Ubiquitous Computing for Cognitive Aids, 2002.
- [55] Charles Antony Richard HOARE.
« Communicating sequential processes ».
Communications of the ACM, 21(8):666–677, 1978.

BIBLIOGRAPHIE

- [56] Arran HOLMES, Hakan DUMAN et Anthony POUNDS-CORNISH.
« The iDorm : Gateway to Heterogeneous Networking Environments ».
In International ITEA Workshop on Virtual Home Environments, 1:20–21,
2002.
- [57] Ian HORROCKS et OTHERS.
« DAML+OIL : A Description Logic for the Semantic Web ».
IEEE Data Eng. Bull., 25(1):4–9, 2002.
- [58] Manon Moal Sylvain Giroux Dominique Lorrain Helene Pigot HUBERT
KENFACK-NGANKAM, Camille Deguy.
« Smart Daily Objects for Aging Well at Home », 2016.
- [59] Larson K. & Tapia E. M. INTILLE, S. S.
« Designing and evaluating technology for independent aging in the home. ».
In International Conference on Aging Disability and Independence., pages 1–15,
2003.
- [60] Stephen S INTILLE.
« Designing a Home ».
Ieee Pervasive Computing, 1(2):76–82, 2002.
- [61] Daniel JACKSON.
Software Abstractions.
MIT press Cambridge, 2 édition, 2006.
- [62] Vikramaditya R JAKKULA, Aaron S CRANDALL et Diane J COOK.
« Knowledge Discovery in Entity Based Smart Environment Resident Data
Using Temporal Relation Based Data Mining ».
In Seventh IEEE International Conference on Data Mining Workshops, pages
625–630, 2007.
- [63] Cliff B JONES.
« Systematic software development using VDM ».
Dans *Prentice Hall International Series in Computer Science*, 1986.
- [64] Georgia M. KAPITSAKI, George N. PREZERAKOS, Nikolaos D. TSELIKAS et
Iakovos S. VENIERIS.

BIBLIOGRAPHIE

- « Context-aware service engineering : A survey ».
Journal of Systems and Software, 82(8):1285–1297, 2009.
- [65] Hubert KENFACK NGANKAM, Hélène PIGOT, Marc FRAPPIER, Camila H OLIVEIRA et Sylvain GIROUX.
« Formal Specification for Ambient Assisted Living Scenarios ».
Dans Sergio F OCHOA, Pritpal SINGH et José BRAVO, éditeurs, *Ubiquitous Computing and Ambient Intelligence*, pages 508–519. Springer International Publishing, 2017.
- [66] Patrick KINNEY et OTHERS.
« Zigbee technology : Wireless control that simply works ».
Dans *Communications design conference*, volume 2, pages 1–7, 2003.
- [67] Lech KRZANIK et Tuomas RITOLA.
« Evaluating DIY smart experience with generative mashups ».
Dans SYSTEMS et 2012 International Conference on INFORMATICS (ICSAI), éditeurs, *2012 International Conference on Systems and Informatics, ICSAI 2012*, pages 2438–2442. IEEE, 2012.
- [68] Ora LASSILA et Ralph R SWICK.
« Resource description framework (RDF) model and syntax specification ».
Citeseer, 1998.
- [69] Clare M. LEWANDOWSKI, New CO-INVESTIGATOR et Clare M. LEWANDOWSKI.
« The effects of brief mindfulness intervention on acute pain experience : An examination of individual difference ».
Revista Guia de la Empresa Familiar, 1:1689–1699, 2015.
- [70] Chin Yang LIN, Tien Yan MA, Che Wei HU, Ting Wei HOU et Fuh Gwo CHEN.
« A DIY approach to creating smart daily services for elders living alone ».
CSAE 2012 - Proceedings, 2012 IEEE International Conference on Computer Science and Automation Engineering, 2:163–167, 2012.
- [71] Adolfo LOZANO-TELLO et Asunción GÓMEZ-PÉREZ.
« Ontometric : A method to choose the appropriate ontology ».
Journal of Database Management (JDM), 15(2):1–18, 2004.

BIBLIOGRAPHIE

- [72] Rajib MALL.
Fundamentals of software engineering.
PHI Learning Pvt. Ltd., 2018.
- [73] Jussi MATTILA et Antti VÄÄTÄNEN.
« UbiPlay : an interactive playground and visual programming tools for children ».
Interaction Design and Children, page 129, 2006.
- [74] Deborah L MCGUINNESS, Frank VAN HARMELEN et OTHERS.
« OWL web ontology language overview ».
W3C recommendation, 10(10):2004, 2004.
- [75] Georgios MEDITSKOS, Stamatia DASIOPOULOU, Vasiliki EFSTATHIOU et Ioannis KOMPATSIARIS.
« Sp-act : A hybrid framework for complex activity recognition combining owl and sparql rules ».
Dans *Pervasive Computing and Communications Workshops (PERCOM Workshops)*, 2013 *IEEE International Conference on*, pages 25–30. IEEE, 2013.
- [76] Prabhakar V MHADSE et Amol C WANI.
« Speaker Identification Based Automation System Through Speech Recognition ».
International Journal of Emerging Technology and Advanced Engineering, 3(1):557–563, 2013.
- [77] Giulio MORI, Fabio PATERNÒ et Carmen SANTORO.
« CTTE : Support for developing and analyzing task models for interactive system design ».
IEEE Transactions on Software Engineering, 28(8):797–813, 2002.
- [78] Natalya F NOY, Deborah L MCGUINNESS et OTHERS.
« Ontology development 101 : A guide to creating your first ontology », 2001.
- [79] M. OLIVARES, S. GIROUX, P. DE LOOR, A. THÉPAUT, H. PIGOT, S. PINARD, C. BOTTARI, G. LE DORZE et N. BIER.
« An ontology model for a context-aware preventive assistance system : reducing exposition of individuals with Traumatic Brain Injury to dangerous situations

BIBLIOGRAPHIE

- during meal preparation ».
- 2nd IET International Conference on Technologies for Active and Assisted Living (TechAAL 2016)*, pages 3 (8 .)–3 (8 .), 2016.
- [80] Camila Helena Souza OLIVEIRA, Sylvain GIROUX, Hubert NGANKAM et Hélène PIGOT.
« Generating Bayesian Network Structures for Self-diagnosis of Sensor Networks in the Context of Ambient Assisted Living for Aging Well ».
Dans *International Conference on Smart Homes and Health Telematics*, pages 198–210. Springer, 2017.
- [81] Jason PASCOE.
« Adding Generic Contextual Capabilities to Wearable Computers ».
International Symposium on Wearable Computers, 44(0):92–99, 1998.
- [82] Theodore PATKOS, Ioannis CHRYSAKIS, Antonis BIKAKIS, Dimitris PLEXOUSAKIS et Grigoris ANTONIOU.
« A reasoning framework for Ambient Intelligence ».
Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 6040 LNAI:213–222, 2010.
- [83] P PENTLAND, Alex.
« Smart Rooms ».
Scientific American, 274(April):54–62, 1996.
- [84] James L PETERSON.
« Petri net theory and the modeling of systems ».
PrenticeHall, inc, Englewood Cliffst1, 981, 1981.
- [85] Yves PUNIE.
« The Future of Ambient Intelligence in Europe :The Need for More Everyday Life ».
Communications & Strategies, 1(57):141–165, 2005.
- [86] Grégoire V Ngankam H. Pigot H. Lorrain D. Giroux RADZISZEWSKI, R.
« Errance nocturne : limiter la déambulation et l’inquiétude de l’entourage ».
CAG, 2016.

BIBLIOGRAPHIE

- [87] Robert RADZISZEWSKI.
« *Système d'assistance lors d'errance nocturne chez les personnes âgées atteintes d'Alzheimer* ».
Thèse de doctorat, Université de Sherbrooke, 2017.
- [88] Robert RADZISZEWSKI, Hubert KENFACK NGANKAM, Hélène PIGOT, Vincent GRÉGOIRE, Dominique LORRAIN et Sylvain GIROUX.
« An Ambient Assisted living nighttime wandering system for elderly ».
The 18th International Conference on Information Integration and Web-based Applications & Services (iiWAS2016), 2016.
- [89] Robert RADZISZEWSKI, Hubert KENFACK NGANKAM, Hélène PIGOT, Vincent GRÉGOIRE, Dominique LORRAIN et Sylvain GIROUX.
« Designing Calm and Non-intrusive Ambient Assisted Living System for Monitoring Nighttime Wanderings ».
International Journal on Perceptive and Cognitive Computing, 1(1), 2016.
- [90] Parisa RASHIDI, Student MEMBER, Diane J COOK, Lawrence B HOLDER, Senior MEMBER et Maureen SCHMITTER-EDGEcombe.
« Discovering Activities to Recognize and Track in a Smart Environment ».
IEEE transactions on knowledge and data engineering, 23(4):527–539, 2011.
- [91] Daniele RIBONI et Claudio BETTINI.
« From lab to life : Fine-grained behavior monitoring in the elderly's home ».
Proc. of PerCom . . ., pages 342–347, 2015.
- [92] Vincent RICQUEBOURG, David MENGA, David DURAND, Bruno MARHIC, Laurent DELAHOUCHE et Christophe LOG ? ?.
« The smart home concept : Our immediate future ».
2006 1st IEEE International Conference on E-Learning in Industrial Electronics, ICELIE, pages 23–28, 2006.
- [93] Marc ROELANDS, Laurence CLAEYS, Marc GODON, Marjan GEERTS, Mohamed Ali FEKI et Lieven TRAPPENIERS.
« Enabling the Masses to Become Creative in Smart Spaces ».
Dans Dieter UCKELMANN, Mark HARRISON et Florian MICHAHELLES, éditeurs,

BIBLIOGRAPHIE

- Architecting the Internet of Things*, pages 37–64. Springer Berlin Heidelberg, 2011.
- [94] Yvonne ROGERS.
« Moving on from Weiser’s Vision of Calm Computing : Engaging UbiComp Experiences ».
Dans Paul DOURISH et Adrian FRIDAY, éditeurs, *UbiComp 2006 : Ubiquitous Computing*, pages 404–421. Springer Berlin Heidelberg, 2006.
- [95] A G SALGUERO et M ESPINILLA.
« Ontology-based feature generation to improve accuracy of activity recognition in smart environments ».
Computers and Electrical Engineering, 68(May 2017):1–13, 2018.
- [96] Maria J. SANTOFIMIA, Scott E. FAHLMAN, Xavier DEL TORO, Francisco MOYA et Juan C. LOPEZ.
« A semantic model for actions and events in ambient intelligence ».
Engineering Applications of Artificial Intelligence, 24(8):1432–1445, 2011.
- [97] Bill SCHILIT, Norman ADAMS et Roy WANT.
« Context-aware computing applications ».
Dans *1994 First Workshop on Mobile Computing Systems and Applications*, pages 85–90, 1994.
- [98] Gary SCOTT et Jeannette CHIN.
« A DIY approach to pervasive computing for the Internet of Things : A smart alarm clock ».
2013 5th Computer Science and Electronic Engineering Conference, CEEC 2013 - Conference Proceedings, pages 57–60, 2013.
- [99] Audrey SERNA, Hélène PIGOT et Vincent RIALLE.
« Modeling the progression of Alzheimer’s disease for cognitive assistance in smart homes ».
User Modelling and User-Adapted Interaction, 17(4):415–438, 2007.
- [100] Graeme SMITH.
The Object-Z specification language.
volume 1. Springer Science & Business Media, 2012.

BIBLIOGRAPHIE

- [101] T. SOHN et A. DEY.
« iCAP : an informal tool for interactive prototyping of context-aware applications ».
Conference on Human Factors in Computing Systems, page 974–975, 2003.
- [102] Alexandru SORICI, Gauthier PICARD, Olivier BOISSIER, Antoine ZIMMERMANN et Adina FLOREA.
« CONSERT : Applying semantic web technologies to context modeling in ambient intelligence ».
Computers and Electrical Engineering, 44:280–306, 2015.
- [103] Jeffrey SPIESS, Yves T JOENS, Raluca DRAGNEA et Peter SPENCER.
« Using Big Data to Improve Customer Experience and Business Performance ».
Bell Labs Technical Journal, 18(4):3–17, 2014.
- [104] J Michael SPIVEY et J R ABRIAL.
The Z notation.
Prentice Hall Hemel Hempstead, 1992.
- [105] Neville A. STANTON.
« Hierarchical task analysis : Developments, applications, and extensions ».
Applied Ergonomics, 37(1):55–79, 2006.
- [106] Robert STEVENS et Margaret STEVENS.
« A Family History Knowledge Base Using OWL 2. ».
Dans *Owled*, volume 432, page 1, 2008.
- [107] Thomas STRANG et Claudia LINNHOFF-POPIEN.
« A Context Modeling Survey ».
Graphical Models, Workshop o(4):1–8, 2004.
- [108] Rudi STUDER, V Richard BENJAMINS, Dieter FENSEL et OTHERS.
« Knowledge engineering : principles and methods ».
Data and knowledge engineering, 25(1):161–198, 1998.
- [109] H. HAGRAS TAWIL, E.
« Adaptive on-line co-ordination of ubiquitous computing devices with multiple objectives and constraints. ».
Dans *In Proceedings of the IEE International Workshop on Intelligent Environments*, pages 116–124, 2005.

BIBLIOGRAPHIE

- [110] Alessandra TONINELLI, Rebecca MONTANARI, Lalana KAGAL et Ora LASSILA.
« A semantic context-aware access control framework for secure collaborations in pervasive computing environments ».
Dans *International semantic web conference*, volume 4273, pages 473–486. Springer, 2006.
- [111] Prof TOYOTA DREAM HOUSE PAPI.
« Sakamura ».
University of Tokyo, Japan, 2004.
- [112] By Federico VIANI, Member IEEE, Fabrizio ROBOL, Alessandro POLO, Paolo ROCCA, Member IEEE, Giacomo OLIVERI, Member IEEE, Andrea MASSA et Member IEEE.
« Wireless Architectures for Heterogeneous Sensing in Smart Home Applications : Concepts and Real Implementation ».
Proceedings of the IEEE, 101(11):2381–2396, 2013.
- [113] Jia WANG et Robert LINDEMAN.
« Coordinated hybrid virtual environments : Seamless interaction contexts for effective virtual reality ».
Computers & Graphics, 48:71–83, 2015.
- [114] X.H. WANG, DA QING ZHANG, TAO GU et H.K. PUNG.
« Ontology based context modeling and reasoning using OWL ».
IEEE Annual Conference on Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second, pages 18–22, 2004.
- [115] M. WEISER.
« Hot topics-ubiquitous computing ».
Computer, 26(10):71–72, 1993.
- [116] Mark WEISER.
« The computer for the 21st century ».
Scientific american, 265(3):94–104, 1991.
- [117] Mark WEISER et John Seely BROWN.
« The Coming Age of Calm Technology ».
Dans *Beyond Calculation : The Next Fifty Years of Computing*, pages 75–85. Springer New York, 1997.

BIBLIOGRAPHIE

- [118] Niklaus WIRTH.
« Program development by stepwise refinement ».
Dans *Pioneers and Their Contributions to Software Engineering*, pages 545–569. Springer, 2001.
- [119] Jong-bum WOO et Youn-kyung LIM.
« User Experience in Do-It-Yourself-Style Smart Homes ».
Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing - UbiComp '15, pages 779–790, 2015.
- [120] Tatsuya YAMAZAKI.
« Beyond the Smart Home ».
2006 International Conference on Hybrid Information Technology, 2:350–355, 2006.